
Octubre 1985-250 ptas.
Todospectrum
AÑO 2 - NUMERO 14
REVISTA EXCLUSIVA PARA USUARIOS
Microdrive, ese desconocido
Para arquitectos: Cálculos de estructura
Listado: Diseñador de pantallas
QL




Portada: Pachecha


Director: Simeón Cruz

# 14

AÑO II

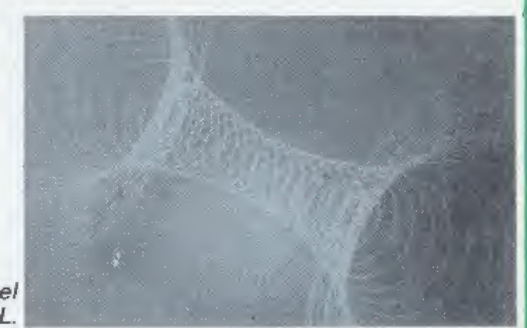





*Cálculo de estructuras: El Spectrum aplicado a la ingeniería.*



*Técnicas avanzadas del microdrive.*



*Dibujando con el GraphiQL.*



*Este número de octubre se abre con un artículo destinado a todos los afortunados poseedores del microdrive. Con los comentarios que hacemos, podréis optimizar el uso de este periférico. También comentamos un paquete de programas comerciales destinados al cálculo de estructuras y que esperamos haga las delicias de los arquitectos. Por último seguimos describiendo el nuevo operativo del Spectrum, dando el listado en ensamblador para aquellos que deseen modificarlo.*

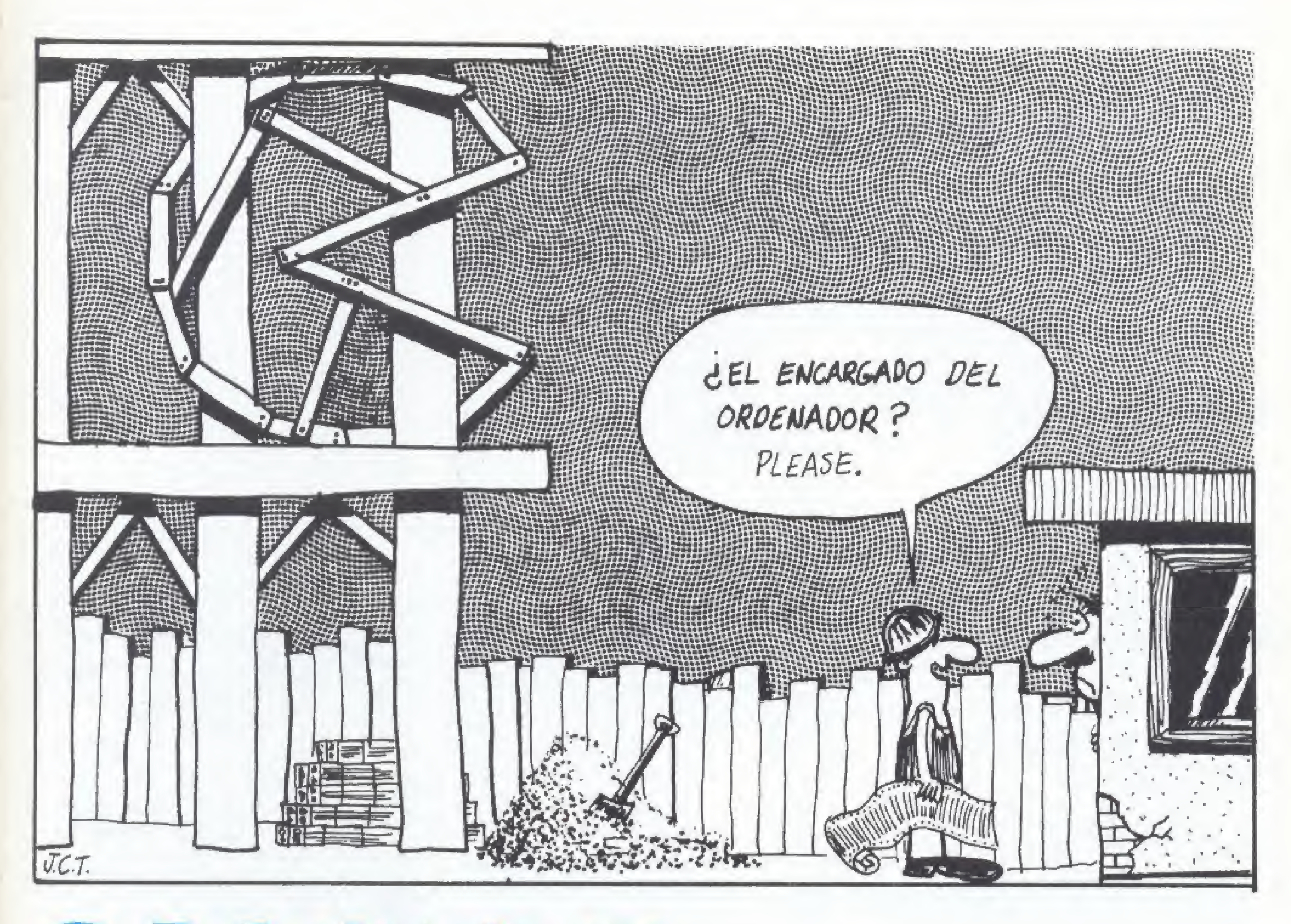
# ACTUALIDAD

```
10  REM EL SPECTRUM 128K YA ES UNA REALIDAD: Este proyecto
        "fantasma" del que tanto se ha hablado ya es una
        realidad. Según nos han confirmado fuentes de
        Investrónica, esta maquina, que todavía no está
        disponible, tendrá entre otras cosas un teclado
        númerico independiente, interfaz MIDI para
        instrumentos musicales y dos modos de operación:
        emulación del Spectrum plus y uno nuevo que sabe
        utilizar la memoria extra como un disco RAM para
        almacenar datos y programas. Pronto ofreceremos más
        noticias.

20  REM LOS ORDENADORES DOMESTICOS PUEDEN SUBIR: Aunque no
        por las razones habituales de la carestía de la
        vida, sino por un extraño arancel "proteccionista"
        que el gobierno ha decidido implantar. Este fija un
        pago mínimo de aduanas de 15000 pesetas para todo
        ordenador que no sobrepase las 64K de memoria. Esto
        puede hacer que muchos fabricantes suban los
        precios o que alguien se ponga a fabricar en
        España. Como va a hacer, casi con toda seguridad,
        Investrónica con el nuevo 128K.

30  REM SOFTWARE PARA EL SPECTRUM 128: Durante la feria
        "Personal Computer World Show" celebrada hace poco
        en Londres, muchas casas de software anunciaban que
        estaban trabajando en programas especiales para el
        nuevo Spectrum. Entre ellas estaban Mikro-Gen y
        OCP.

40  REM MANIA PINTORA: Al igual que durante algún tiempo
        abundaron los juegos olímpicos, lo que está de moda
        ahora son los programas de dibujo similares al
        "MacPaint" del ordenador Macintosh de Apple. La
        mayoría de estos programas podían usar
        opcionalmente un "ratón" de una marca u otra.
```




# CALCULO DE ESTRUCTURAS

*¿Qué pensarían ustedes si les dijeran que su vivienda hubiera sido calculada con un Spectrum 48K? No tienen por qué asustarse, la colección de programas de Paraninfo Soft que vamos a comentar permite realizar estos cálculos en por lo menos un 90 por 100 de los casos y con resultados más que correctos*

El Spectrum es un ordenador personal, aunque para muchos no pasa de ser un juguete que tiene muchas posibilidades. Es común el pensar que un Spectrum sirve sólo para jugar y que para realizar trabajos serios debemos recurrir a ordenadores profesionales que se diseñan exclusivamente para eso. Así, si un arquitecto cree que necesita un ordenador para su estudio, la posibilidad de que el ordenador Spectrum sea el escogido está prácticamente descartada, siendo lo normal el buscarlo entre los productos que ofrecen las multinacionales del sector, mucho más completos, rápidos y caros.

Cuando se va a adquirir un ordenador profesional es normal que el *hardware,* o máquina, sea mucho más barato que el *software* o programas. Ello es esencialmente porque en este tipo de mercado es más importante lo que hace que con que lo hace.

Pero no debemos olvidar que hace muy pocos años un ordenador con la capacidad y potencia del Spectrum, no sólo hubiera sido profesional sino que hubiera sido de lo más cotizado y además los paquetes de programas de cálculo de estructuras no son un invento reciente. Luero no nos debe extrañar que con un Spectrum se puedan hacer cálculos profesionales serios aunque tampoco hay que engañarse, los ordenadores profesionales actuales son mucho más completos y efectivos.

Si usted es un arquitecto y considera que no tiene el suficiente trabajo para rentabilizar un ordenador profesional, puede que tenga en su Spectrum una solución razonable, e incluso puede que le interese comprar el Spectrum y el paquete de programas de **Paraninfo Soft** para destinarlo a un uso profesional.

Para los lectores no versados en estos temas vamos a tratar de explicar de una manera sencilla y en un lenguaje asequible para qué se usan estos programas. Un cálculo de estructuras consiste básicamente en la determinación de las dimensiones y características de los elementos constructivos para obtener la estructura deseada.

Sí, por ejemplo, queremos analizar cómo se hace una vivienda, hay que distinguir dos fases, la de diseño y la de cálculo. En la primera se debe plasmar en un dibujo cómo queremos que sea la vivienda y qué usos vamos a darle, en esta fase se incluye la parte artística de la arquitectura.

Una vivienda se compone de una estructura que es la que soporta el peso de la misma, y de unos cerramientos que son las paredes y techos, aunque esta distinción no sea muy rigurosa vale para andar por casa. La estructura es algo así como el esqueleto de la edificación, nosotros sabemos dónde están los huesos, que se llaman vigas si son horizontales y pilares o columnas si son verticales, y las articulaciones, llamadas nudos y que son los puntos donde concurren dos o más vigas y pilares. Lo que queremos saber es cuál debe ser el tamaño de las vigas, pilares y cimientos, para que la estructura aguante y sea segura ante adversidades desconocidas mediante los coeficientes de seguridad.

La colección de programas vienen ordenados secuencialmente para calcular, a partir de los pesos máximos previstos, cómo se reparten los esfuerzos sobre la estructura y a partir de ello se deduce la dimensión que tienen que tener los elementos constructivos para poder soportar esos esfuerzos. Todos estos métodos de cálculo están normalizados, es decir, que deben ser realizados siguiendo esquemas de cálculo redactados por las autoridades competentes y que como puede comprobarse pueden ser implementados en el Spectrum.

Nos damos cuenta que para un lector profano en la materia los comentarios de los programas pueden resultar un poco incomprensibles, pero creemos que la profesionalidad que demuestran tener estos programas les hace merecedores de ser incluidos en una revista divulgativa aunque no técnica. El Spectrum puede hacer mucho más de lo que se cree vulgarmente.




## La colección

Cuando nos pusimos a realizar el comentario de esta colección imaginamos otra cosa, supusimos




***Cálculo de Estructuras* software *básico para los arquitectos***

que serían unos programas básicos sacados de algún curso de Resistencia de Materiales o de Estructuras meramente educativo y sin apenas utilidad práctica. Al cargarlos en el ordenador cambió totalmente nuestra impresión, el programa era muchísimo más serio y pedimos la colaboración de un arquitecto en ejercicio que no creyó lo que le estábamos contando hasta que no lo vio, y mucho nos tememos que esto sea lo que le pase a usted. De todas maneras sepa que los programas han sido escritos por José Luis del Río Moreno, de **Paraninfo Soft, S. A.,** que había trabajado también en los programas de cálculo de estructuras de **Hewlett Packard.**

La colección de programas se compone de dos secciones, Estructuras de Hormigón y Estructuras Metálicas.

La colección de Estructuras de Hormigón se compone de 5 programas.

—Memoria de cálculo y cargas.
—Cálculo de esfuerzos en pórticos.
—Cálculo de armaduras de vigas.
—Cálculo de armaduras de pilares.
—Cálculo de cimientos.

La colección de Estructuras Metálicas se compone de 4 programas.

—Memoria de cálculo y cargas.
—Cálculo de perfiles en vigas.
—Cálculo de perfiles en pilares.
—Cálculo de cimientos.

Estos programas se presentan lujosamente encuadernados, con dos versiones una para impresora ZX y la otra para impresora Centronics. También se suministra una cinta *cassette* virgen de 10 minutos que no he llegado a comprender el por qué de su presencia. Los folletos de instrucciones no son excesivamente completos aunque en la ejecución de los programas se repiten las instrucciones, suponiendo una pesadez cuando los programas ya han sido ejecutados varias veces.

La introducción de los datos es la adecuada, con suficientes interpelaciones verificatorias, que desgraciadamente son necesarias. Resulta imprescindible el trabajar con impresora, debido a la cantidad de datos que son necesarios y también a la cantidad de datos que se obtienen de los programas. El único «defectillo» apreciable en la introducción de datos no es que nos podamos salir de rango en los valores, sino que escribamos más decimales de los normales estropeando la presentación del resultado en la pantalla, pero para un usuario profesional esto no tiene sentido.

La presentación es agradable, tal vez tenga excesivos colorines, que suponemos habrán realizado así aprovechando las posibilidades del Spectrum, pero que puede ejercer una impresión negativa en algunos profesinales que confunden lo serio con lo sobrio. Afortunadamente la informática moderna con su nuevo desarrollo de gráficos interactivo tiende a que cada vez su programas presenten un aspecto visual más agradable.

## Descripción de los programas

Como los programas van en series cuasiparalelas y para no hacer una crítica pormenorizada de nueve programas el comentario va a ser común para ambas colecciones.

El primer programa de la colección es la Memoria de Cálculo y Cargas. En su inicio nos ofrece la muestra de los valores típicos sobre los que va a trabajar, resultando estos valores los normales en construcción, admitiéndose además una gama de valores más que suficiente que, por ejemplo, llega hasta admitir una presión admisible sobre el terreno de hasta 100 kg/cm². El programa nos pregunta si hay terrazas, si son voladas, y si hay alguna planta con sobrecarga. Definimos la altura de la cinco plantas, el tipo de cubierta, tipo de fachada, escalera y si hay ascensor.

El programa sólo nos da dos opciones en cuanto a cubierta inclinada: teja y pizarra. En este aspecto es pobre, pues no considera teja de cemento o árabe, cubierta de fi-

brocemento, de paneles *sandwich* de chapa, de *«shingles asfáltica»*...

Referente a las sobrecargas, el programa sólo da opción a que exista una planta con sobrecarga distinta de la general de plantas, lo cual resulta un poco pobre para edificios en que se agrupan diferentes usos.

Las opciones relativas al doblado de las fábricas son pobres también, pues sólo considera tabique, tabicón o carencia de doblado, olvidando tecnologías muy en boga como el yeso cartón y también la escayola.

El programa no pide sobrecarga de uso en terraza o nieve, poniendo en total 200 kg/m$^2$, que es una cifra conservadora.

El juicio general es que se trata de un buen programa que sirve para que un arquitecto redacte su preceptiva Memoria de Cálculo de Estructura de un Proyecto, para visar en el Colegio Profesional y entregar al cliente. Los procedimientos de cálculo son los normalizados.

Los defectos que se pueden señalar son los siguientes:

—Limitación de opciones sobre las que escoger en algunos casos, que contrastan con la gran amplitud en otros.

—El apartado de cálculo de zapatas que se redacta en la memoria así como en aquel en el que se habla del método de cálculo, parece más una clase de estructuras que un documento de un proyecto.

La calificación que merece este programa es excelente.




## Cálculo de esfuerzos en pórticos

El siguiente programa rompe la simetría entre ambas colecciones, ya que sólo se presenta para estructuras de hormigón.

Este programa es el que suele caracterizar a la potencia de todo el paquete, ya que es el más crítico, por lo que creo resultará más interesante al profesional que hablemos aquí de las posibilidades que permite.

Inicialmente nos presenta los valores típicos característicos que son los mismos que en el programa anterior. Se nos pregunta posteriormente cuál es el número de nudos de la estructura teniendo como límite superior 80. Después hace lo mismo con las vigas estando el máximo en 80 y el mínimo en el número de nudos. En el programa aparecen otra vez instrucciones sobre los tipos de pilares, ya que el programa no pide número de pilares, sino tipos diferentes de pilares, en cuanto a sección o altura. Estos pilares deben ser predefinidos, no ocurriendo esto con las vigas. En las instrucciones se nos invita a que realicemos un esquema previo de la estructura, numerando las vigas de acuerdo a una norma muy simple y denominando a los nudos en función de su situación en la estructura de acuerdo a una ordenación que también se muestra en el programa, así como la nomenclatura de vigas y pilares que concurren a un nudo, resultando bastante chocante que a las vigas que concurren a un nudo se las llame Bde y Biz. Una vez introducida la topología de la estructura nos llevamos la sorpresa de que han variado las instrucciones sin cambiar el librito, lo cual parece un poco chapucero, este cambio consiste en que hay que dar las cargas ya calculadas introduciendo un peso aproximado de la viga, con lo que se eliminan errores en la introducción de los datos. Así se introducen los datos de la viga, luz y cargas y se obtiene los valores de los cortantes a cada lado y los momentos a cada lado y en el centro. Nos pide que fijamos un valor para la sección de viga, h ó b dándonos como margen entre 15 y 120 cm; si no nos complace la sección de viga tenemos dos opciones, o fijamos otro nuevo valor o introducimos la sección aparte del programa.

El final del programa nos ofrece el cálculo de los momentos hiperestáticos, dándonos además otras constantes de cálculo, que pueden servir como comprobación, que son la rigidez y el coeficiente de reparto. También nos calcula los esfuerzos finales en vigas dando el momento isostático en el centro y los momentos hiperestáticos y cortantes isostáticos en los extremos, y concluye con los momentos finales en los pilares.

Este programa es brillante y nos hace recordar los terribles esfuerzos que realizaron generaciones de técnicos para hacerse a mano el reparto de esfuerzos mediante el método de Cross, operaciones que el ordenador nos soluciona rapidísi-

mamente y «cerrando todos los nudos», es decir igualando perfectamente los esfuerzos (con dos decimales), cosa casi imposible o por lo menos muy difícil de realizar manualmente.

La crítica más seria que puede hacerse de este programa es que a nuestro juicio han mal utilizado memoria operativa dentro del programa escribiendo instrucciones completamente innecesarias si se hubiera redactado un manual de instrucciones claro. Además debemos añadir que dichas instrucciones son evidentes sobre todo después de haber ejecutado el programa varias veces y sólo sirven para que el usuario teclee la «s» para pedir el cambio de pantalla. Siguiendo por aquí me parece poco imaginativo el usar la «s» para contestar afirmativamente a preguntas y pasar páginas, el *«autorepeat»* nos puede jugar alguna mala pasada, este defecto ocurre en toda la colección.

## Otros cálculos

Los siguientes programas son los del cálculo de vigas, de los perfiles en estructuras metálicas y de las armaduras en las de hormigón. Estos programas calculan todo a partir de las normas usuales en construcción. El programa de Cálculo de Perfiles nos da a escoger entre perfiles IPN, IPE o ambos. Los datos más interesantes que hay que introducir son: n.º de plantas, n.º de vigas, y pilares por planta, la topología de la estructura (aquí son estéreas), la carga y la luz de cada viga. Obteniendo el programa para cada viga, los cortantes sobre los lados, el momento en el centro, el tipo de perfil, la tensión de trabajo, las flecha y su relación longitud flecha y el peso de la viga. Además nos da el peso de vigas por planta, las cargas en los pilares por planta y las cargas totales acumuladas. La capacidad de cálculo de este programa abarca 176 vigas.

Para el programa de cálculo de armaduras en vigas, el programa pide un diámetro para armar, si es demasiado el programa probara con armaduras de tamaño menor, hasta el 2 ø 10. Si el diámetro es mayor aumentará el de las barras para poder armar todo en una capa. El programa pide características del pórtico, n.º de plantas y vigas a calcular. Para cada viga ncs pide un número de datos: canto, ancho, luz, cortantes en los lados, momentos en los extremos y en el centro. Después nos pide la topología de la estructura. Los resultados que nos ofrece el programa son: Arranques izquierdo y derecho calculando las armaduras superior e inferior, lo mismo en el centro del vano, así como la armadura a cortante. También presenta los cortantes hiperestáticos y para armar así como los estribos en los arranques. Finalmente nos da las cargas verticales en pilares, en cada planta y totales por planta.

En la entrada de esfuerzos no permite corregir el primer dato, sino tan sólo el segundo. Hay que considerar que sólo se deben meter las vigas diferentes que existen a armar, no todas las existentes, pues aunque sea evidente, se tiende a introducir todas las vigas existentes.

Al no haber gráficos en el programa, no podemos definir la longitud del armado, por lo que hay que completarlo con un cálculo manual, lo que es verdaderamente un fallo, aunque excusable porque no debemos olvidar que estamos trabajando con un Spectrum.

Lo siguiente es el cálculo de armaduras de pilares en estructuras de hormigón y cálculo de perfiles de pilares en estructuras metálicas.

El programa de cálculo de armaduras nos solicita la definición de la estructura, plantas y pilares por planta, la dimensión de los pilares, las cargas y momentos por plantas. El programa nos ofrece el cálculo de las cargas totales acumuladas por planta. Las armaduras las calcula ofreciendo además una serie de datos adicionales como son: dimensiones, carga vertical, momento en los ejes X e Y, excentricidad según los mismos ejes, excentricidad adicional y total según los ejes. Y por supuesto las armaduras, con la carga que soportan y los estribos.

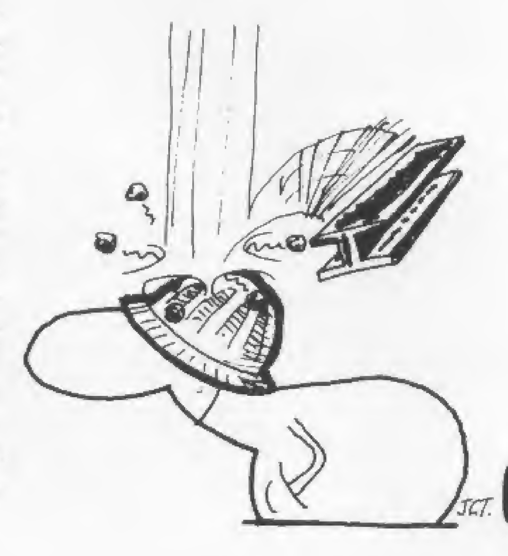
El programa de cálculo de perfiles en pilares nos da a escoger entre tres tipos de pilares: 2-UPN soldadas a tope, 2-UPN empresilladas, y perfil HEB. Después de definir la estructura y las cargas, momentos y altura de cada pilar, el programa ofrece los resultados que son: carga vertical, momentos según los ejes, altura del pilar, perfil del pilar, momentos de inercia, radios de giro, momentos resistentes, coeficiente de pandeo, sección, tensión de trabajo y peso del pilar, además nos da el peso total por planta y el peso total.

Este programa presupone que entramos con valores calculados previamente, como todos los anteriores el trabajo de introducción de datos se multiplica enormemente por la independencia de los programas. Lo más espectacular reside en que el programa contiene todos los datos de los perfiles HEB, UPN soldadas y empresilladas.

Los programas de Cálculo de cimientos son lo más simple de la colección aparentemente, su funcionamiento es correcto y los resultados que nos ofrece son: tipo de zapata, pilares centrado o descentrados según una o dos direcciones. Cargas y momentos sobre la zapata, excentricidades, dimensiones de las zapatas, armaduras.

## Comentario general

La crítica general más razonable que puede hacerse a esta colección



de programas es que no hayan intentado realizar una colección compacta, que tuviera solución de continuidad con la ejecución de los diferentes programas, así una vez cargado el primero el programa almacenará los datos oportunos para que no tuvieran que ser introducidos en el siguiente. También resulta inconveniente mostrar las instrucciones dentro del programa y sobre todo tan repetidamente como se hace aquí, creo que se debe considerar al usuario lo suficientemente inteligente como para suponer que puede trabajar con el manual al lado y que pueda comprender todo, ya que con poca experiencia en el manejo de estos se puede adquirir un control del programa bastante positivo. Los manuales de instrucciones son un poco deficientes.

La solución de continuidad es complicada, sobre todo teniendo en cuenta que trabajamos con programas grabados en *cassette* y con un ordenador pequeño, además probablemente motivos económicos indujeron a **Paraninfo Soft, S. A.,** a separar los programas independizándolos porque tal vez el precio de una colección de programas no debe superar el precio de un microordenador. Esperemos que esta creencia cambie, ya que a nuestro juicio vale más la propia capacidad de los programas que muchas de las máquinas.

**Cesar Hernández**

Tenemos que agradecer la colaboración prestada por el arquitecto en ejercicio D. Juan Luis Mateo Jiménez, que ha comentado conmigo estos programas y que ha realizado una impresión general sobre la colección.

«La primera sorpresa al trabajar con los programas es ver que dan muchos más resultados y datos de los que el usuario de un microordenador puede esperar.

Los programas pueden cubrir las necesidades de un estudio de Arquitectura en general, aunque existe la necesidad de apoyar los programas con algunos cálculos manuales y dibujos exteriores.

Pero lo más impresionantes es la relación calidad-precio. Por unas 60.000 ptas. se consigue un apoyo estructural importante, que cubre la mayoría de los casos que ocurren en edificación, a excepción de muros.

Para muchos sera más interesante comprarse estos programas y un Spectrum, en vez de pagar el programa de estructuras del ordenador profesional que posean. Otro punto interesante es la rapidez. Con un ordenador con capacidad limitada parecía lógico pensar que los cálculos y elección de perfiles



durará un cierto tiempo, y son prácticamente instantáneos. El resumen es que son unos programas francamente interesantes y a un precio asequible».

El resumen es que estamos ante una auténtica colección Profesional excelente.




# HELP

**Periódicamente aparecen programas destinados a facilitarnos la programación en BASIC, con comandos y opciones que faltan en el Spectrum. En este caso es** Dimensión New **quien nos presenta un producto de este tipo. La versión que comentamos viene en** micro-drive**, lo que mejora su velocidad de carga.**

En el cartucho aparecen dos copias, una ejecutable con RUN y otra, sin autoejecución, que se puede cargar con LOAD, aunque está pensada para su uso con MERGE, como veremos más adelante. El programa principal está escrito en BASIC y en el mismo cartucho hay una serie de subrutina, a las que se accede desde un menú cargándose las adecuadas, con la ventaja que supone al no ocupar mucha memoria. En este programa principal nos aparece un menú inicial que nos indica las posibilidades que tiene el programa: ocho útiles rutinas localizables a partir de 64000. Dentro de esta serie de rutinas tenemos desde un renumerador de líneas a un mapa de memoria. La lista de rutinas es:

—Renumerador de líneas (A). Funciona bastante bien a excepción de saltos con expresiones variables como puede ser:

GOTO 1000*S

imposibles de renumerar automáticamente. En este caso el programa nos indica que se ha encontrado una expresión no renumerable. Se puede especificar el número de la primera línea, así como el incremento entre líneas.

—Elimina sentencias REM, que durante la programación previa nos han podido servir como indicativos (B). Revisa línea por línea, siendo un buen sistema para ahorrar memoria en los programas finalizados.

—Mapa de memoria (C). Es una rutina muy útil, que no es necesario cargar, ya que está en BASIC, dentro del programa principal del *Help*. Nos indica la memoria que ocupa el programa, las variables y gráficos definidos, además de la memoria aún disponible.

—Borrador de bloques de líneas (B). Util cuando el número de líneas a borrar es grande.

—Compactador de programa (C). En el Spectrum, difícilmente resulta muy útil, ya que los espacios no hace falta ponerlos. Revisa todo el programa, quitando todos los códigos de control, así como los espacios innecesarios ahorrando memoria.

—Cambia instrucciones (F). Busca en el programa las instrucciones o letras que queramos y las cambia por las deseadas. Puede ser útil en casos especiales como adaptar programas de pantalla a impresoras cambiando PRINT POR LPRINT o cambiar el nombre a alguna variable.

—Listado de variables (G). Puede ser interesante para programas largos, ya que nos muestra una lista de todas las variables, numéricas y alfanuméricas de nuestro programa, incluyendo las variables dimensionadas. Pero que no se asuste nadie, pueden aparecer las variables X$, V$, W$, W y V que son las del *Help*, que permanecen a pesar de que, como veremos, el programa desaparece al ejecutarse la rutina.

—Variables a cero (H). Actúa poniendo a cero las variables que se hayan definido, a diferencia de CLEAR, que las borra.

—Salida (I). Va directamente al comando STOP del siguiente menú.

Una vez que se elige la subrutina, se carga desde el *microdrive* la deseada. La ordenación corresponde al nombre que poseen en el cartucho.

Una vez cargada la rutina pasamos al segundo menú, en el que se nos indica las posibilidades de cargar el programa a depurar desde *microdrive* o *cassette*, ejecutar la rutina deseada, volver al menú o parar la sesión. Hay que

hacer notar que aquí tiene el programa su punto más flaco, ya que borra las líneas del programa Help, para lo cual tarda casi dos minutos, que es, en nuestra opinión, un tiempo excesivo.

Posteriormente se ejecutará nuestra rutina, que generalmente no tarda mucho a no ser que le digamos que renumere muchas líneas o el programa sea largo.

En cuanto a nuestro programa se puede decir que habrá que tener algunas precauciones. No debe contener líneas comprendidas entre 9950 y 9999, así como tampoco una variable v$() o llenar toda la memoria, ya que hay que dejar del orden de 5K para el código máquina y el programa. No obstante, podemos disminuir los problemas cargando sólo la rutina en código máquina a partir de 64000.

```
HELP PROGRAMACION © DIMensionNEW

            OPCIONES

A.-Renumerador.   E.-Compac Prog
B.-Elimina REM.   F.-Cambia Com.
C.-Mapa Memoria.  G.-Listar Var.
D.-Borra lineas.  H.-Var. a Cero

            I.-Salida.

        indica una opcion
```

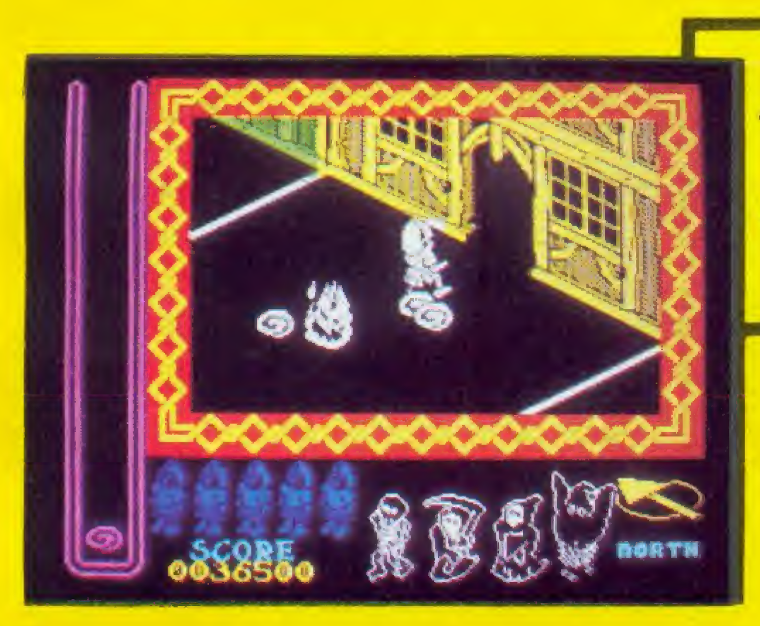
*Pantalla informativa del programa*

En síntesis, es un programa con algunos puntos flojos que pueden servir para «decorar» o depurar nuestros programas ya hechos. Su utilidad como ayuda a la programación es menor, aunque algunas de las opciones pueden ser interesantes.

# NIGHTSHADE

ABC Soft
Spectrum 48 K
2.100 ptas.

De una ciudad apacible, y tranquila, Nightshade pasó a ser un lugar sombrío y desolado donde imperan las tinieblas y las fuerzas del mal sembraron con sus maleficios la oscuridad. Todos aquellos temerarios exploradores que se adentraron en la ciudad para desentrañar su misterio no volvieron jamás. Sin embargo, a pesar de todos los peligros que aguardan tras sus muros tu aventurero se dispone a atravesar la ciudad y destruir los demonios y fantasmas que la habitan, para ello cuenta con cinco vidas, en cada una de ellas puede tener hasta tres contactos con los demonios, cambiando a un color diferente cada vez, primero amarillo, después verde y al tercer contacto muere.




*Nightshade, comienza la aventura contra los cuatro enemigos de la humanidad (representados en la parte inferior de la pantalla).*

Cuando el aventurero atraviesa una puerta o gira sobre una casa, los muros de esta desaparecen, quedando en su lugar una línea blanca uniforme que no podrá ser atravesada a no ser que sea discontinua, en este caso indicaría que hay una puerta. Los muros desaparecen para que podamos ver los movimientos del explorador al entrar en las casas.

El objetivo del aventurero es, como debemos suponer, encontrar y destruir los demonios que, están representados en la parte inferior de la pantalla y, una vez que acabe con ellos, desaparecerán las tinieblas y toda la ciudad pasará a su poder.

En cuanto a los gráficos exhibidos hay que decir que se han reali-




*Estamos en una ciudad llena de fantasmas, pero a diferencia de otros juegos de Ultimate, en este podemos disparar a los enemigos.*

**Control:** El juego puede funcionar tanto por teclado como por joystick.
**Jugadores:** Uno.
**Gráficos:** Excelentes gráficos en 3 dimensiones.
**Nivel de dificultad:** Depende de las situaciones; en general, se mantiene la dificultad a lo largo de todo el juego.
**Originalidad:** Reside principalmente en la combinación de las 3 dimensiones con la buena definición de los gráficos.
**Conclusión:** Es un juego interesante por la dificultad que encierra conseguir el objetivo final. En un principio los porcentajes que se obtienen del juego son bastante bajos.

zado mediante un novedoso proceso llamado «Filmation II», en tres dimensiones, que consigue crear una perspectiva más realista en los movimientos y dan una verdadera sensación de participación en el juego; destacando, además, los gráficos de las casas y los fantasmas que son bastante originales.

# HYPERSPORTS

ERBE Software
Spectrum 48 K
2.100 ptas.

Como hemos podido observar día a día el deporte gana nuevos adeptos y, por ello, no es de extrañar que el mercado del *software* reciba continuamente nuevos juegos cada vez más atractivos sobre los más variados deportes.

En realidad este juego tiene ya un fuerte precedente en versiones similares aparecidas en videojuegos y en otros desarrollados también para Spectrum, aunque las pruebas que incluye son diferentes: natación, tiro al plato, potro, tiro con arco, triple salto y levantamiento de peso.

Cuanto más deprisa se pulsen las teclas de control mayor será el impulso que reciba el deportista y más posibilidades tendrá de superar la marca. A los usuarios del Spectrum normal (con teclado de goma) quizás les resulte un poco más difícil que a los del Spectrum plus.

La puntuación que consigues se muestra en la parte superior izquierda de la pantalla, mientras que los records de cada especialidad obtenidos hasta el momento aparecen en la parte superior derecha.

Los gráficos son de gran calidad dando una buena definición y visión del atleta en todos sus movimientos.

*Tiro al plato. Una de las pruebas más divertidas... ¡hasta que se falla!*




*Salto al potro, una prueba como otra cualquiera para acabar de romper su joystick.*

**Control:** Teclado o joystick.
**Jugadores:** Uno.
**Gráficos:** Muy buenos, destacando los movimientos de los atletas.
**Sonido:** Bastante conseguido, sobre todo los aplausos del público.
**Nivel de dificultad:** La dificultad es creciente a medida que se superan las pruebas.
**Originalidad:** La amplia gama de juegos deportivos resta originalidad a éste, lo que no es motivo para que no sea interesante.
**Conclusión:** La competitividad es el aliciente de este juego, ya que almacena los records conseguidos por cada participante.



*Muchos de los aficionados que se animan a comprar un microdrive, lo hacen buscando la comodidad que supone el uso de estos aparatitos. Y, obviamente, todos desean almacenar en ellos los programas que más utilizan —juegos principalmente—. La desilusión llega en seguida al darse cuenta aun en el caso de que fuera posible usar este cómodo sistema, no se sabe cómo.*

# MICRODRIVE, el gran desconocido

En el presente artículo no se exponen técnicas maravillosas ni remedios infalibles. De hecho, hay muchos programas que no pueden almacenarse en *microdrive.* Pero es seguro que con lo que se comenta, muchos van a ser capaces de meter en cintura alguno de sus programas rebeldes.

Además se darán una serie de consejos de uso general para los usuarios de *MICRODRIVE,* EL GRAN DESCONOCIDO.

## Conversión de formato

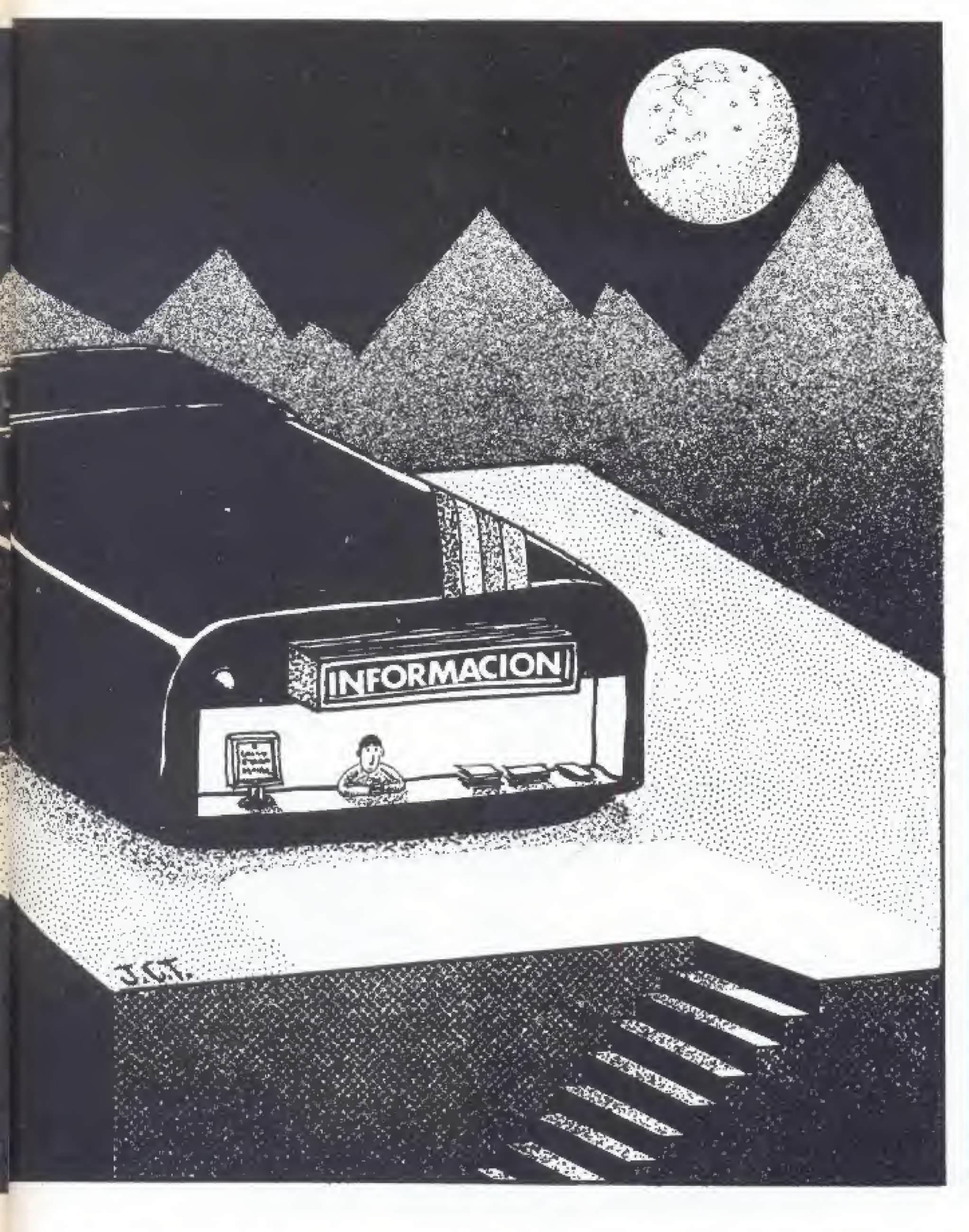
Son varios los problemas que se presentan al aficionado a la hora de realizar la conversión *cassette-microdrive,* pero podemos dividirlos en dos grandes grupos: los referentes a la memoria disponible y los relacionados con el cambio medio, que supone la alteración de las instrucciones de carga.

El hecho de conectar un *Interface* I supone que el usuario contará con 79 *bytes* menos de RAM. En efecto, para el correcto funcionamiento del mencionado *interface* —sin el que los *microdrives* no pueden funcionar—, es necesario ampliar el número de variables del sistema. Realmente 58 *bytes* no son muchos (un 0,16 por ciento de la memoria de un Spectrum 48 K). Pero aquí no acaba la cosa, pues esto es sólo del *interface.* Si hace uso de un *microdrive,* el sistema necesita otros 595 *bytes* por cada canal abierto. Recordemos que un LOAD, SAVE,... supone la apertura y cierre automático de un canal. De estos, 512 *bytes* corresponden al *buffer* de datos. (En la jerga informática, el término *buffer* corresponde a un área de memoria destinada al intercambio de información entre entre periféricos de distintas velocidades).

Por tanto, en el mejor de los casos, por ejemplo el escribir un LOAD * «m»; 1; «xxxx» supone que:

— se dispondrá de 79 + 595 = 674 *bytes* menos que si se trabajara con cinta de *cassette.*

— no se podrá trabajar en el es-




«Bytes», el tema se presenta algo más complicado, pero algo puede hacerse. En el siguiente apartado encontrará información sobre el tema.

En el caso mencionado, cargue el primer bloque haciendo uso del comando MERGE. De esta manera evitará que el programa se autoejecute al entrar, y así podrá echarle un vistazo. Entre otras cosas, encontrará una o varias líneas de programa donde se indica la carga del resto de los bloques

```
Bloque numero 1
Nombre: Monsters
Longitud: 109
No es un CODE

Bloque numero 2
Nombre: 1
Longitud: 6912
Comienzo: 16384

Bloque numero 3
Nombre: 1
Longitud: 9471
Comienzo: 23296
```

pacio comprendido entre las direcciones 23734 y 24408.

Estas dos formas de «ver la vida» corresponde a una única realidad, pero dan fe de los dos problemas que se plantean: la cantidad de memoria disponible y el lugar donde se dispone.

### Memoria: Cantidad disponible

El porcentaje de la memoria sustraída sigue siendo pequeño (1,3 por ciento con 48 K de RAM). Sin embargo, esto puede crearnos problemas en algunos casos.

Todos los «maníacos cazamarcianos» saben que hay programas —los mejores—, que ocupan hasta el último *byte* de la memoria disponible. Evidentemente, un programa de este tipo, que ha sido pensado para funcionar con cinta magnética, no podrá ser nunca almacenado en *microdrive* ya qe el funcionamiento .de ambos —programa y *microdrive*— requieren una cantidad de memoria que el ordenador no dispone.

Pero, ¿cómo saber si el programa que nos ocupa necesita o no más memoria de la que podemos darle?

Típicamente un programa comercial consta de varios bloques, el primero de los cuales suele estar en BASIC y se encarga de gestionar la entrada del resto. Si el programa fuera de un solo bloque del tipo (LOAD, a saber). También verá ,posiblemente un CLEAR x. Si el número x es más bajo que 24610, los problemas que se plantean son mucho más serios, y deberá usar las estrategias del siguiente apartado. La prueba más fiable que pueda hacerse es ver si el CLEAR hace referencia a un RAMTOP por encima de PEEK 23641 +256*PEEK 23642 + 595, antes de que realicemos todos los cambios pertinentes.

Porque, en efecto, hay que hacer cambios. En primer lugar se deberá modifr las instrucciones LOAD “”CODE por otras más apropiadas al nuevo entorno. Tome nota cuidadosamente de los posibles números que pudiera haber después del CODE. Se refieren a las direc-

# OFERTA DE SUSCRIPCION

## Todospectrum

REVISTA EXCLUSIVA PARA USUARIOS

*Te ofrece la posibilidad de suscribirte con unas condiciones muy ventajosas para ti:*

**1** *Recibir puntualmente, en tu domicilio la publicación **TODOSPECTRUM que durante 12 meses te proporcionará lectura, programas, ayuda, entretenimiento, etc.***

**2** *Consigue un práctico regalo:*




*Una obra fundamental en la biblioteca de los aficionados al Spectrum:*

- *Reglas y herramientas del Basic.*
- *La técnica de los organigramas.*
- *El fabuloso mundo de las rutinas.*
- *Variables y cadenas.*
- *Funciones matemáticas usuales.*

***Esta publicación, escrita con estilo ameno y práctico, te ayudará a sacar todo el partido posible a tu máquina.***

**3** *La opción de ser protagonista. Tú puedes tener una participación directa con tus comentarios, programas, sugerencias, etc.*

**4** *Obtener premios importantes con tus programas, y temas de interés.*

## EN DEFINITIVA, TODO SON VENTAJAS

*No dejes pasar esta oportunidad, suscríbete a "TODOSPECTRUM", cumplimentando hoy mismo el cupón de respuesta adjunto.*



# YA ESTÁ A LA VENTA




## PIDALA EN SU QUIOSCO HABITUAL O SOLICITELA A:

## INFODIS, S.A.

**Bravo Murillo, 377 • 28020 Madrid**

VIRGIN-1-SPECTRUM

**CUPON DE PEDIDO**

Ruego me envien la cinta cossette **VIRGIN SOFTWARE** al precio de **525** pts.
El importe lo abonaré: Con mi tarjeta de crédito ☐ American Express ☐
Visa ☐ Interbank ☐
Contra reembolso ☐ Adjunto cheque ☐
Número de mi tarjeta ________
Fecha de caducidad ________
NOMBRE ________
DIRECCION ________
CIUDAD ________
C.P. ________ PROVINCIA ________
Sin gastos de envio

ciones donde deben introducirse los programas. Caso de existir, son indicio claro de que la posición en la que entraría el programa es otra, lo que requiere especial atención.

Pero bien pudiera ocurrir que después de hacer las modificaciones, el ordenador nos de un error del tipo *«RAMTOP not good»* o *«Out of Memory»*. En efecto, los cambios supondrán casi siempre un aumento de la capacidad de memoria que ocupa el programa BASIC. Y esto, unido a lo que necesita el *microdrive*... Pero no desespere. El problema puede tener solución: vamos a comentar algunos trucos que usados en el programa BASIC, permiten reducir memoria. Este tipo de apaños son extremadamente populares entre los usuarios del ZX-81 versión 1 K.

Los más efectivos son los que eliminan los números. Así, si usa mucho un cierto valor, como tal vez «255», resulta conveniente convertirlo en una variable. Los números poco usados deberán convertirse en cadenas. Asi RANDOMIZE USR xxxxx, puede escribirse como RANDOMIZE USR VAL «xxxxx». De este modo ahorraremos 3 *bytes*. Otras estrategias de asignación pueden verse en la tabla 1.

**Tabla I**

| N.º | Equivalente | Bytes ahorrados |
|---|---|---|
| 0 | NOT PI | 4 |
| 1 | SGN PI | 5 |
| 3 | INT PI | 5 |
| n | VAL "n" | 3 |
| n<255 | CODE "caract ascii" | 2 mínimo |

```
 10 DEF FN B(W)=PEEK W+256*PEEK (W+1)
 20 LET SUM=0:LET IX=23296:LET RUT=2340
    0
 30 FOR I=0 TO 14:READ A:POKE RUT+I,A:L
    ET SUM=SUM+A:NEXT I
 40 IF SUM<> 1396 THEN PRINT "ERROR en
    el DATA":BEEP 1,40: STOP
 50 CLS :LET N=0
100 LET N=N+1:RANDOMIZE USR RUT:PRINT '
    INVERSE 1;"Bloque numero ";N
110 PRINT "Nombre: ";:FOR I=1 TO 10:PRI
    NT CHR$ PEEK (I+IX);:NEXT I:PRINT

120 PRINT "Longitud: ";FN B(IX+11)
130 IF PEEK IX<> 3 THEN PRINT "No es un
     CODE":GO TO 100
140 PRINT "Comienzo: ";FN B(IX+13):GO T
    O 100
150 DATA 221,33,0,91,175,55,17,17,0,205
    ,86,5,48,242,201
160 REM
170 REM RUT LD   IX,5B00H
180 REM     XOR  A;Cabecera
190 REM     SCF   ;LOAD
200 REM     LD   DE,17;Longit
210 REM     CALL 0556H
220 REM     JR   NC,RUT
230 REM     RET
```

El lector se preguntará por qué tenemos tanto interés en eliminar los números. Los números son voraces respecto a la memoria que necesitan. Sólo por existir, ocupan 6 *bytes*. A esto hay que añadirle un *byte* por cada cifra que posea. Así el «0», ocupa 7 *bytes*, y el «0000», 10 *bytes*. De los 6 *bytes* indicados, uno es el «código de número» (14d, 0EH) y los otros cinco, su equivalente en el formato de punto flotante. Más información sobre esto, en el capítulo 24 del Manual del Usuario.

Otra recomendación en la misma línea es el uso de variables de una sola letra o de variables en los GOTO o GOSUB.

Volvemos al asunto que teníamos pendiente.

Al usar el programa alguna vez, habrá observado si alguna de las partes es un dibujo de presentación. Caso de ser así, téngalo en cuenta para no grabarlo. Por una parte, le ahorrará una memoria sustancial. Por otra, dada la velocidad de los *microdrives*, no va a tener mucha oportunidad de admirarla.

A continuación, numere los bloques «útiles» y asigne a cada uno un nombre. Es una buena idea que el nombre sea precisamente el número asignado, pues son preferibles los nombres cortos. Vamos a comenzar nuestro trabajo. Cargue en primer lugar el programa lector de cabeceras que proponemos (prog. 1). Anote las longitudes y

comienzo de los bloques que van apareciendo. Elimine de la lista los que no considere útiles (dibujo, instrucciones...). Una vez acabada la labor, borre el programa. A continuación escriba CLEAR xxxxx, donde xxxxx es el número que aparece con el CLEAR del programa que va a copiar.

Repase la hoja donde apuntó la distribución de las instrucciones de carga. Si la primera es LOAD ""CODE, escriba esto mismo. Si fuera LOAD ""CODE xxxxx, hágalo igualmente. Cualquier otra posibilidad se resuelve del mismo modo. Ponga en marcha la cinta y espere a que finalice la carga. Escriba ahora SAVE *«m»; 1; «1» CODE com, long donde long es la longitud anunciada por el programa lector de cabeceras y com es la dirección de comienzo del programa y será la indicada en el CODE de carga, o más corrientemente en su defecto el mostrado por el programa 1. El número 1 hace referencia al orden, aunque el nombre puede ser cualquiera, preferiblemente corto.

Repita la operación tantas veces como partes tenga el programa.

Cargue de nuevo el programa BASIC de gestión, pues es el turno de modificarlo. Deberá sustituir todos los LOAD por LOAD *«m»; 1;«nom» CODE, donde nom es el nombre dado a cada sección (un número según la recomendación hecha). Si hay muchos bloques, puede ser interesante hacer uso de un FOR-NEXT.

Grabe este programa de la forma SAVE *«M»; 1; «nom» LINE 1, donde nom es el nombre del programa, lógicamente el mismo de la versión *cassette*. Si todo ha ido bien podrá echarse una partidita. En otro caso, revise todos los pasos, tal vez haya cometido un error, o no haya tenido en cuenta algún detalle.

### Memoria: posición

Como ya adelatábamos, los problemas pueden venir no por limitación de la memoria disponible, sino porque el *Microdrive* y el *interface* necesitan la misma zona de memoria que el programa. Lamentablemente, de hecho, este es el caso más frecuente. La solución es más delicada y requiere una cierta labor de programación en código máquina. Es el caso típico de programas que cargan de un solo bloque el dibujo de presentación, el programa BASIC y el programa en código máquina. Veamos cómo se resolvió el problema para un divertido juego: «MONSTERS IN HELL».

El programa tiene tres bloques, de los cuales eliminamos el central, que es un dibujo de presentación. El tercer bloque lo grabamos en posiciones más altas de memoria, concretamente en 45056:

LOAD ""CODE 45056
SAVE *"m"; 1; "MONS. COD." CODE 45056,9471




```
1 PAPER 0:INK 0:BORDER 0:CLS
2 CLEAR 45000:LOAD *"m";1;"Mons.COD"C
  ODE
3 FOR I=0 TO 11:READ A:POKE 45056+I,A
  :NEXT I:RANDOMIZE USR 45056
4 SAVE *"m";1;"MONSTERS"LINE 1
5 DATA 1,255,36,17,0,91,33,0,176,237,
  176,201
```

Ahora retocamos el programa y los grabaremos igualmente. Es el que hemos llamado Programa 2.

Al cargarse, dará el mensaje de error *«Out of Memory»*. Pero no tema, bastará con hacer GOTO 4 para solucionar el problema.

***El tiempo de acceso a la información se reduce considerablemente***

## Consejos

**Nombres:** Un problema que presenta el CAT del *microdrive* frente a otros sistemas de almacenamiento en disco (o incluso la cinta del Spectrum) es que nos ofrece información sobre el tipo de archivo que se trae entre manos. Para conservar esta útil posibilidad, podemos introducirla de forma implícita en el nombre. Si usamos *tokens*, podemos determinar inmediatamente el tipo de archivo reservando para ello un único *byte* (carácter) del nombre. Así un programa en código máquina puede caracterizarse por el *token* «CODE», las matrices con «DATA». Los programas requerirán más letras: «.BSC» para BASIC, o «.TXT» para los textos. Así el programa de cabeceras puede llamarse «CABEZ.BSC». Las restricciones en la longitud del nombre compensan con creces la información que proporciona.

**CAT con 40 títulos:** El comando CAT no es capaz de dar más de 20 nombres antes de que aparezca el mensaje de *«scroll?»*. El programa que proponemos (prog. 3) eleva este número a 40, con la contrapartida de usar un sector de *microdrive,* que borrará de forma automática al acabar.

**Reducción del tiempo de acceso:** Para la reducción del tiempo de acceso, hay dos posibles estrategias a seguir. Una no requiere el empleo de mucha materia gris, y por tanto no es óptima. Se trata de hacer varias copias del programa en cuestión dentro del mismo cartucho. Esto se realizará cómodamente del siguiente modo:

POKE 23791, n de copias: SAVE *"m"...

El POKE se hace sobre la variable del sistema COPIES, que guarda el número de copias a realizar. El valor usual es 1.

El motivo de la rapidez se debe a que, si bien el *microdrive* tiene una velocidad de transferencia bastante alta, el tiempo de acceso deja bastante que desear. Reduciendo el tiempo que pasa hasta que se localizan los datos, habremos ganado bastante tiempo del total.

El segúndo método requiere algo más de elaboración, pero no supo-

```
 10 REM    CAT ampliado
 20 REM
 30 POKE 23658,8:REM Mayusculas
 40 LET M$="Usamos la impresora? (S/N)"
   :GO SUB 1000
 50 IF D$<> "S" AND D$<> "N" THEN BEEP
   .1,40:GO TO 40
 60 LET P$=D$
 70 LET M$="Drive no>>":GO SUB 1000
 80 LET D=VAL D$
 90 CLOSE # 10
100 OPEN # 10;"M";D;"?"
110 CAT #10;D:REM CAT al drivE
120 PRINT #10;CHR$ 0;CHR$ 0:REM Fin del
    CAT
130 CLOSE # 10
140 OPEN # 10;"M";D;"?"
150 LET A$=" "
160 LET A$=A$+INKEY$ #10:LET J=LEN A$
170 IF A$(J)<> CHR$ 0 OR A$(J-1)<> CHR$
    0 THEN GO TO 160
180 CLOSE # 10:ERASE "M";D;"?"
190 LET S=2:IF P$="S" THEN LET S=3:REM
   Canal de salida
200 FOR A=1 TO J-2
210 IF A$(A)<> CHR$ 12 THEN PRINT #S;A$
   (A);:GO TO 240
220 IF A$(A+1)=CHR$ 13 THEN PRINT #S:PR
   INT #S,:GO TO 240
230 PRINT #S,
240 NEXT A
250 PRINT TAB 10;"Drive:";D: STOP
000 INPUT (M$);D$:IF D$="" THEN GO TO 1
   000
010 RETURN
```

ne ningún tipo de contrapartida. Sin embargo, sólo podrá usarse en cartuchos que tengan previamente almacenados pocos programas. Cuando se aplica, puede reducir significativamente el tiempo de acceso en programas de varios bloques. Y lo que puede ser más importante, alarga la vida de las cintas, al reducir el intervalo de funcionamiento de éstas.

Simplemente pretendemos que la carga de los programas se realice inmediatamente uno detrás de otro, sin que sea necesario dar una vuelta completa de búsqueda. Al acabar de grabar el primer bloque, la inercia del motor sigue haciendo girar a la cinta un cierto tiempo (aproximadamente un cuarto de su longitud). Al efectuar una grabación, se hace necesario dar una vuelta completa a la cinta para buscar los sectores libres. El programa se empezará a grabar en los primeros espacios vacíos que se encuentren, esto es, a partir del punto donde se paró. Si nosotros fuéramos capaces de hacer girar el cartucho de modo que la cabeza se posicionara justo después del bloque a continuación de otro. Como se comentaba, el método es sólo efectivo si usamos cintas vacías, ya que cuando usamos cartuchos con varios programas, la asignación se hace según el espacio libre con el que se cuente.

***Es posible obtener catálogos con 40 títulos***

**Varios** *Microdrive:* Aquellos usuarios que disponen de más de un *microdrive*, habrán notado alguna vez que si cargan programas en BASIC que se auto-ejecutan, y el cartucho se halla en una unidad distinta de la que se ha grabado, el proceso acaba siempre en un desastre. Esto es debido a que la instrucción de carga hace necesariamente referencia a una unidad, que puede ser otra distinta de la que está funcionando en ese instante. Para solucionar el problema, deberá usarse la siguiente rutinilla:

LET D=PEEK 23766: REM no del microdrive en función LOAD *"M"; D; "nombre"...

El PEEK proporciona el número de unidad que está funcionado en ese momento, y por tanto, la que tiene el cartucho con el programa que se ejecuta.

**Luis M. Brugarolas**



# Aprendiendo Lenguaje Máquina

# CAPITULO: 3

En meses anteriores ya dimos las ideas básicas de funcionamiento del microprocesador Z-80 y aprendimos unos conocimientos imprescindibles como son los de convertir de decimal a hexadecimal, el sistema binario, la estructura interna del microprocesador, etc.

En este artículo vamos a ver algunas de las instrucciones del ensamblador y su uso, para lo cual haremos algunos programas sencillos donde se ve cómo usarlas.

## Una sencilla suma

Nuestro primer programa va a realizar una suma de dos *bytes* y nos va a dejar el resultado en una posición de memoria de modo que luego podamos verlo por medio de la instrucción PEEK del BASIC.

Para ello vamos a usar dos instrucciones: LD y ADD. La primera es una abreviatura de «LOAD» que significa literalmente «cargar», su uso básico es el de mover un dato de un sitio a otro, o dicho de otro modo, cargar una posición de memoria con el contenido de otra. Esta instrucción tiene multitud de modos distintos de uso, pero en todos ellos detrás del nombre de la instrucción se dan dos parámetros. El primero indica el lugar donde se va a almacenar el dato y el segundo dice de dónde se saca. Por tanto, si decimos LD xx,yy; estamos haciendo que el dato contenido en yy se saque y se introduzca en xx. Todos los tipos de direccionamiento existentes que indicamos en capítulos anteriores se pueden usar con esta instrucción, aunque no se permiten todas las mezclas. En este capítulo y posteriores veremos los que están permitidos.

Leer en A el primer sumando desde 60500.



Sumar a A el segundo sumando (60501).



Guardar el resultado en 60502.



volver al BASIC

FIGURA 1

La otra instrucción que hemos comentado es ADD (suma). Esta, como su propio nombre indica, suma de dos números de un *byte*. Del mismo modo que en la anterior, aquí también hay que indicar de dónde se sacan los dos números a sumar y se puede decir lo mismo respecto a los modos de direccionamiento y restricciones existentes.

Una vez hechas estas indicaciones, hagamos un esquema básico del funcionamiento del programa. Este esquema nos servirá de base para realizar posteriormente diversas variantes en las que veremos distintos modos de direccionamiento y explicaremos sus ventajas e inconvenientes.

El organigrama básico se da en la figura 1. Como se ve su sencillez es extrema y se pueden dividir en tres pasos principales:

— Coger el primer número a sumar.

— Sumarle el segundo.

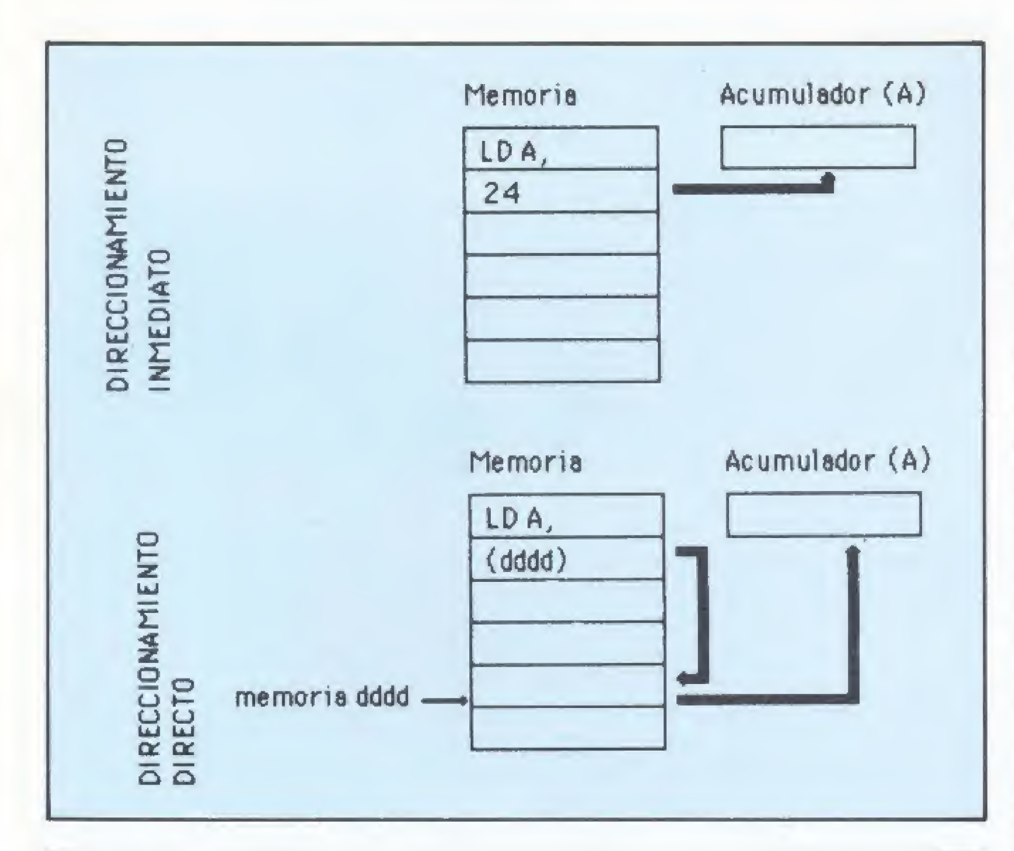
— Guardar el resultado en la dirección de memoria indicada.

Aunque todas las operaciones se podrían hacer en algunos casos en varios de los registros de Z-80, vamos a usar el Acumulador (A) para irnos acostumbrando a usarlo, ya




que es el más versátil y el que más posibilidades ofrece.

Supongamos que los dos números a sumar son 127 (7Fh) y 42 (2Ah). El sistema más sencillo y rápido (el tiempo que tarda el microprocesador en hacerlo) de sumarlos es utilizar el direccionamiento inmediato. Como se recordará, en este tipo de direccionamiento el número que se lee es el que se indica en la misma instrucción. De modo que la primera orden sería "LD A,127". Que hace que se carge el número 127 en el acumulador. Con lo que el primer paso ya está realizado. La segunda instrucción a dar es la que hace que se sume 42 al número almacenado en el acumulador. Para ello utilizaremos otra vez el direccionamiento inmediato y haremos "ADD A,42". Que realiza esta suma y deja el resultado en el mismo acumulador. Por último, hay que guardar este resultado en una memoria para que al volver al BASIC podamos recuperarlo. Esto se hace con la misma instrucción LD, pero cambiando el modo de uso, ya que ahora será "LD (dddd),A". Esto indica que el contenido del acumulador se almacena en la memoria indicada por dddd (que debe ser un número entre 0 y 65535). Existe también (y la usaremos posteriormente) esta misma instrucción, pero con los parámetros cambiados. Es decir "LD A,(dddd)" que saca el contenido de la memoria dddd y lo mete en el acumulador. La diferencia entre ambos tipos de direccionamiento se muestra en la figura 2. En 2a se ve el direccionamiento inmediato, que coge el valor inmediatamente posterior a la instrucción y lo guarda en el acumulador. En 2b se ve cómo este valor (considerado como 16 *bits*) se usa para direccionar una memoria, cuyo contenido es el que se mete en el acumulador. En el caso que estamos tratando usamos, como ya hemos dicho, "LD (dddd), A", debiendo susti-




```
EA60           2     ORG    60000
EA60 3E7F     10     LD     A,127
EA62 C62A     20     ADD    A,42
EA64 3254EC   30     LD     (60500),A
EA67 C9       40     RET
EA68          50     END
```

```
EA60           2     ORG    60000
EA60 3A54EC   10     LD     A,(60500)
EA63 47       20     LD     B,A
EA64 3A55EC   30     LD     A,(60501)
EA67 80       40     ADD    A,B
EA68 3256EC   50     LD     (60502),A
EA6B C9       60     RET
EA6C          80     END
```

tuir dddd por una dirección de memoria.

Con esto se acaba el programa de suma, por lo que añadimos como última instrucción una «RET», que explicaremos posteriormente. Ahora basta saber que hace que el control se devuelva al BASIC al acabar el programa. Si no la pusiésemos perderíamos el control del microprocesador y el ordenador se nos colgaría, por lo que es imprescindible ponerla siempre al final de nuestro programa.

El listado completo se muestra en la figura 3, en la que se da el listado producido por el ensamblador usado por nosotros (otros ensambladores pueden dar resultados ligeramente diferentes) y que da tres columnas. En la primera se ve la dirección donde se meten los códigos generados (y, por tanto, donde va el programa), en la segunda se ven dichos códigos en hexadecimal y en la tercera las instrucciones en ensamblador que hemos metido y que se corresponden a dichos códigos. En este listado se puede ver que el programa se ha puesto a partir de la dirección 60000 (EA60h) por medio de un «ORG 60000». Esto no es una instrucción del Z-80, sino que le dice al microprocesador que empiece a meter las instrucciones a partir de dicha dirección. También se ve que el resultado se almacena en la memoria 60502, aunque podía ser en cualquier otra que no use el Spectrum ni nuestro programa, ya que entonces al guardar el dato borraría lo que hubiese antes, pudiendo producirse resultados inesperados y bastante desagradables en general.

Una vez hecho esto es conveniente salvarlo en cinta (por lo que pueda pasar) y volverlo a cargar cuando queramos ejecutarlo (el método de salvarlo en cinta es distinto para cada ensamblador, por lo que es conveniente leerse las instrucciones. Aquellos que no dispongan de esta utilidad pueden introducir los *bytes* de la segunda columna del listado por medio de un cargador hexadecimal en BASIC, de los que ya han aparecido varios en la revista, y salvarlo con SAVE""CODE dddd,1111", donde dddd es la dirección inicial y 1111 la longitud de los *bytes* a salvar. En primer lugar, y ya desde BASIC, haremos un CLEAR 59999. Esta instrucción le indica al Spectrum que desde esa dirección en adelante el espacio está reservado y que no lo puede usar. Si no la pusiésemos, el ordenador se creería que tiene toda la memoria para él y podría escribir encima de nuestro programa destruyéndolo. A continuación se carga el programa con LOAD""CODE y una vez en memoria se ejecuta haciendo RANDOMIZE USR 60000. Después de teclear esto el cursor volverá a aparecer instantáneamente (si todo ha ido bien), y aunque parezca mentira, se habrá ejecutado el programa y el resultado estará en la dirección 60502, cosa que podemos comprobar haciendo un PRINT PEEK (60502), que nos mostrará el número 169.

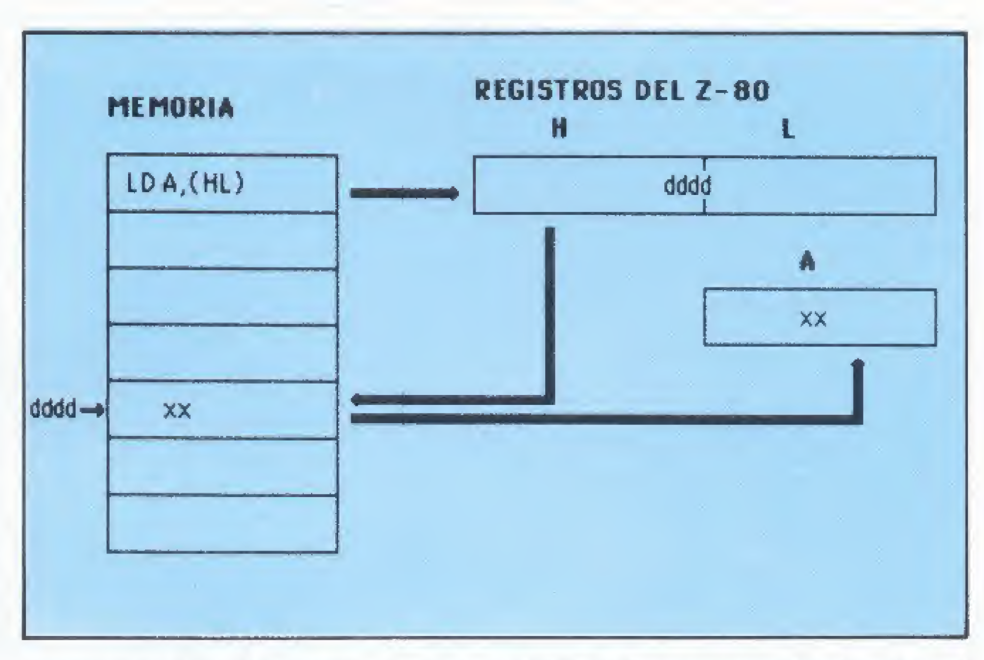
## Otro método más versátil

El programa, como hemos visto, funciona correctamente, pero plantea algunos problemas, ya que si queremos sumar dos números distintos a los indicados, habrá que volver a escribir el programa, ensamblarlo, etc. Evidentemente esto no resulta práctico, por lo que hay que buscar otro sistema.

Para evitar que suceda esto, conviene separar el programa de los datos, es decir, que el programa no tenga los datos como parte componente, sino que los lea de otra zona de memoria (a partir de la 60500, por ejemplo). Para ello se puede usar la instrucción "LD A,(dddd)" que vimos antes, y en este caso dddd debe indicar dónde está al-

| INSTRUCCION | | REGISTROS | |
|---|---|---|---|
| LD A,(60500) | A | 127 | ← |
| | B | xxxxxx | |
| LD B,A | A | 127 | ] |
| | B | 127 | |
| LD A,(60501) | A | 42 | ← |
| | B | 127 | |
| ADD A,B | A | 169 | + |
| | B | 127 | ] |

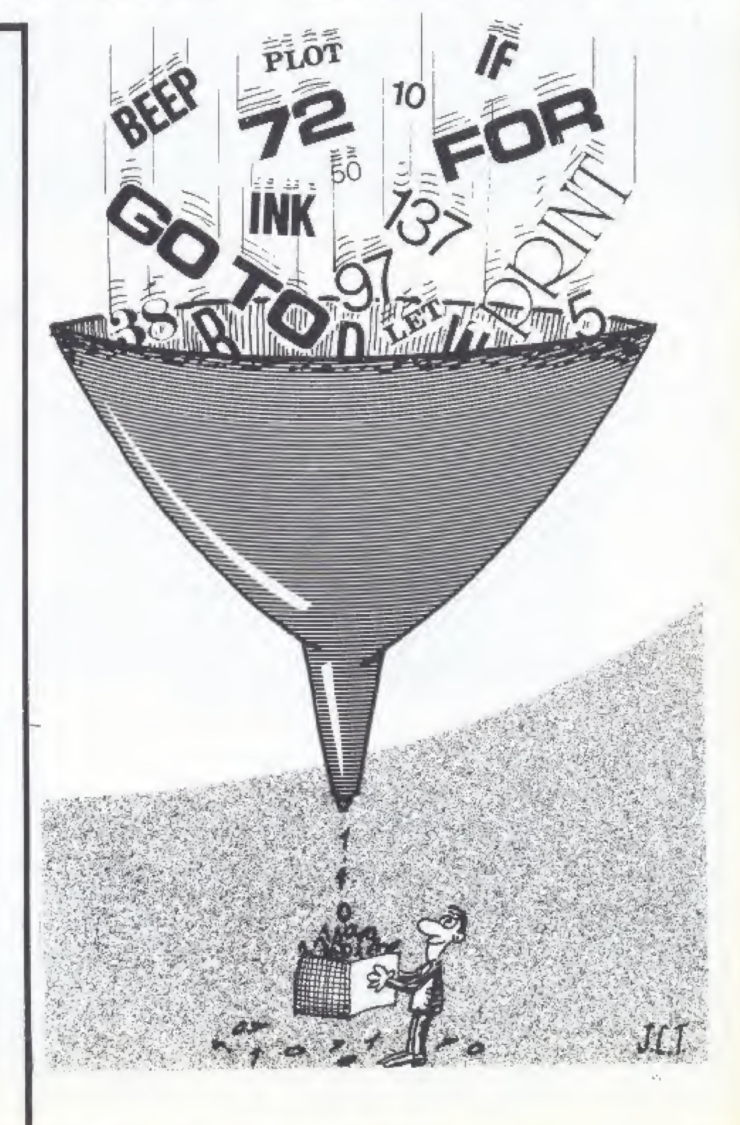
macenado el dato. Por ejemplo, supongamos que en la memoria 60500 está el primer número a sumar, en la 60501 el segundo y el resultado lo almacenamos en la 60502 como antes. Por tanto, el proceso a realizar queda como sigue: En primer lugar leer el número almacenado en la memoria 60500 y guardarlo en el acumulador. A continuación se le debe sumar el contenido de la memoria 60501. Pero el problema es que NO existe una instrucción "ADD A,(dddd)", que nos sumaría al acumulador el contenido de la dirección dddd, lo que nos obliga a utilizar otro sistema.




Este otro sistema puede ser el pasar el contenido del Acumulador (el primer número) al registro B, meter entonces en A el segundo número y sumar A y B. Este proceso se complica ligeramente, pero nos permite tener los datos separados del programa. Debido a este cambio de funcionamiento hay que cambiar dos instrucciones. La primera es "LD B,A", que pasa el contenido de A a B. La segunda es ADD A,B que suma A y B y deja el resultado en A. La parte final es como en el listado anterior y el resultado se vuelve a almacenar en 60502. El listado completo se da en la figura 4.

Para entender mejor el funcio-





namiento del proceso de suma se puede ver la figura 5, en la que se han representado gráficamente los registros internos del Z-80 según se van ejecutando las instrucciones. Como se puede ver, se cargan los datos en A y B y luego se suman. De este modo los datos se pueden cambiar fácilmente. Para ello, antes de ejecutar el programa (por medio de RANDOMIZE USR 60000) hay que introducirlos por medio de POKE 60500,xx con el primer número y POKE 60501,yy con el segundo. El resto del proceso de funcionamiento es igual al primer programa que hicimos.

## Direccionamiento con HL

Este es el tercer tipo de direccionamiento que podemos usar para nuestra suma. Las ventajas que nos reporta son interesantes debido a que nos permite realizar todas las operaciones utilizando como referencia una dirección introducida previamente en HL. A este tipo de direccionamiento se le denomina indirecto, ya que no se da el dato (como en el modo inmediato) ni la dirección de donde cogerlo (como en el direccionamiento directo), sino que se dice de dónde se tiene que coger la dirección que a su vez contiene el dato. En la figura 6 se ve cómo al leer la instrucción el microprocesador, se le dice que saque el contenido del par de registros HL y lo use como otra dirección. Por decirlo de otro modo, se da una dirección (en este caso se indica un registro) que a su vez contiene la dirección de donde sacar el dato.

De este modo podemos realizar las operaciones de carga y suma, ya que el Z-80 sí las admite haciendo referencia a HL, con el único condicionante de que al principio del programa hay que cargar en este par de registros la dirección donde se encuentran los datos. Para ello se recurre a una nueva variante del LD, que se escribe de la forma "LD HL,dddd", siendo dddd (la dirección de los datos) un número de 16 *bits* almacenado en las dos memorias siguientes a la instrucción. La razón de que sea de 16 bits es que esta longitud que tiene HL considerado como un solo registro en vez de dos de 8 *bits* (todos los LD que hemos hecho antes eran de 8 *bits*, ya que éste es el tamaño de las memorias y de los registros A y B del microprocesador). El valor que introduciremos en HL será 60500.

```
EA60            10     ORG     60000
EA60  2154EC    20     LD      HL,60500
EA63  7E        30     LD      A,(HL)
EA64  23        40     INC     HL
EA65  86        50     ADD     A,(HL)
EA66  23        60     INC     HL
EA67  77        70     LD      (HL),A
EA68  C9        80     RET
EA69            90     END
```

```
EA60                 ORG    60000
EA60 DD2154EC        LD     IX,60500
EA64 DD7E00          LD     A,(IX+0)
EA67 DD8601          ADD    A,(IX+1)
EA6A DD7702          LD     (IX+2),A
EA6D C9              RET
EA6E                 END
```

Una vez hecho esto podemos pasar a realizar el cálculo propiamente dicho. En primer lugar se lee el primer dato, que se encuentra en la memoria 60500, por lo que basta hacer "LD A,(HL)" y se nos carga automáticamente el contenido de dicha memoria, ya que está apuntada por HL. Para sumar el segundo dato hemos de hacer "ADD A,(HL)", pero este dato está en la memoria 60501, mientras que según hemos visto HL apunta a 60500, por lo que habrá que incrementarla previamente de modo que apunte a este segundo dato; esto se hace de un modo muy sencillo, ya que hay una instrucción específica del Z-80 que hace exactamente lo que queremos; incrementar HL en uno. Esta instrucción se llama, evidentemente, INC HL. Una vez ejecutada ésta podemos realizar la suma.

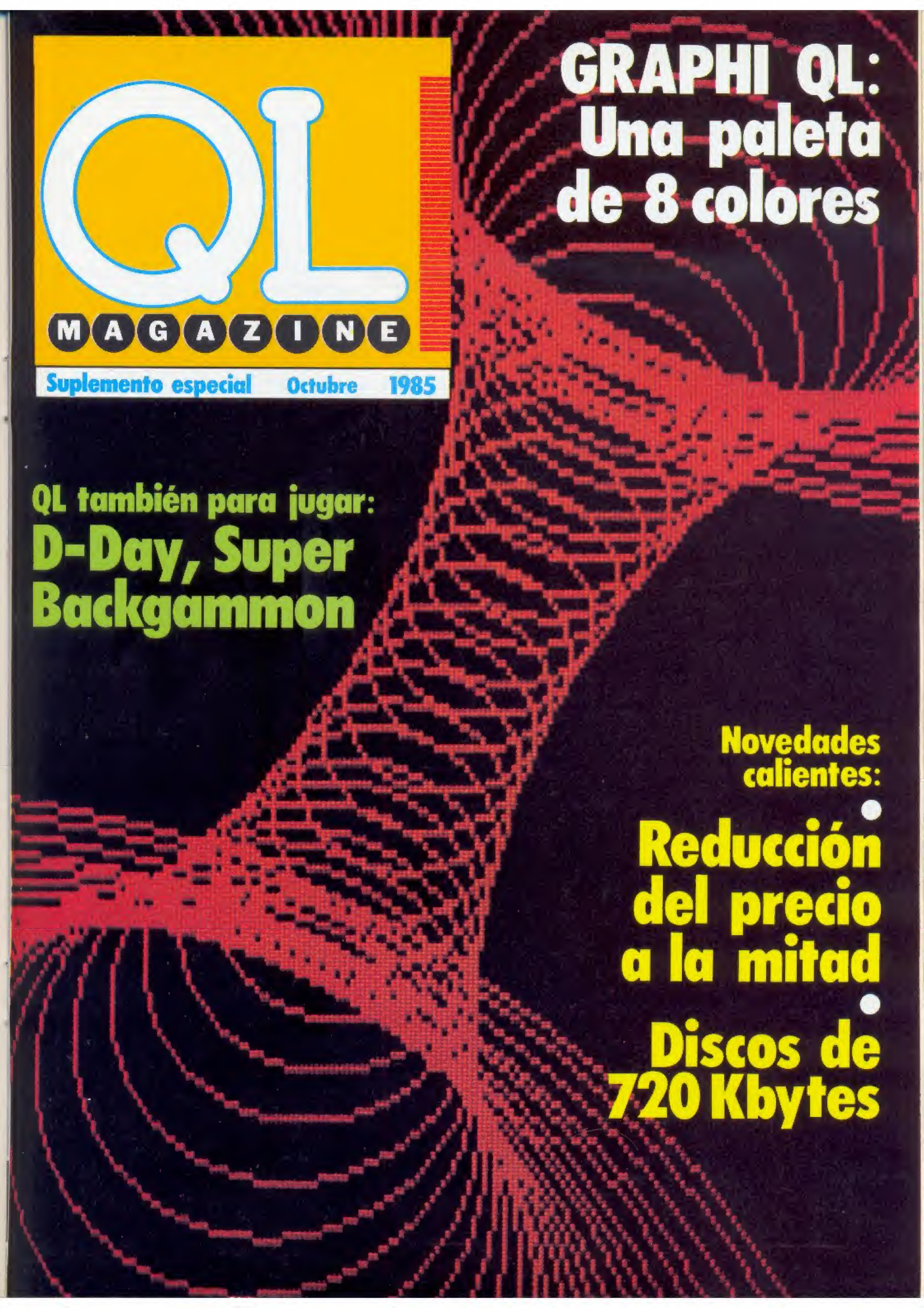
Por último, para guardar el resultado podríamos hacerlo como en las versiones anteriores del programa, pero en su lugar vamos a utilizar también HL. En primer lugar lo incrementamos de nuevo haciendo que señale a 60502 (el lugar donde queremos almacenar el dato) y a continuación hacemos "LD (HL),A" de funcionamiento similar a otras veces, pero en este caso no se indica la dirección, sino que se halla indicada por HL. Para terminar se pone como es habitual el "RET". En la figura 7 se muestra el listado resultante. El uso es igual que en caso anterior (ya que sólo se ha modificado la estructura interna del programa, pero no el modo en que entran y salen los datos) y los datos se introducen haciendo POKEs en 60500 y 60501 y el resultado se saca con un PEEK de 60502.

## El direccionamiento indexado

Este es el último tipo de direccionamiento que vamos a estudiar como aplicación práctica de nuestro programa de suma. Su funcionamiento es muy similar al indirecto que hemos visto con HL, pero con dos diferencias principales. La primera es que en lugar de utilizar el par de registros HL, vamos a usar los IX (también se podrían usar los IY) y la segunda es que el Z-80 nos permite que en la misma instrucción se le sume al contenido de este registro un número de un *byte* para hallar la dirección definitiva. Es muy importante tener en cuenta que esta suma sólo se hace para hallar la dirección y no se modifica el contenido de IX. Es decir, si hacemos "LD A,(IX+2)" y en IX hay un 60500, se nos cargará en el acumulador el contenido de la memoria 60502 (60500+2), pero en IX seguirá estando 60500, de modo que lo podemos usar posteriormente para leer o escribir otras direcciones de memoria. Por tanto, el proceso es similar al anterior, pero simplificado, ya que no hace falta incrementar IX debido a que el desplazamiento se incluye en la instrucción. El programa se lista en la figura 8, en la que podemos ver cómo el primer dato se lee con "LD A,(IX+0)" al estar en la misma memoria 60500, el segundo dato se suma con "ADD A,(IX+1)" para que se lea la dirección 60501 y el resultado se guarda con "LD (IX+2),A" de un modo similar al programa anterior, pero incluyendo el desplazamiento.

Con esto quedan explicados los distintos modos de direccionamiento y su utilización práctica en este caso. El mes que viene veremos como se puede realizar restas y sumas de números mayores por medio del uso de la bandera de acarreo.

**Fernando García**



QL
MAGAZINE
Suplemento especial Octubre 1985
GRAPHI QL: Una paleta de 8 colores
QL también para jugar:
D-Day, Super Backgammon
Novedades calientes:
Reducción del precio a la mitad
Discos de 720 Kbytes

NOTICIAS

## El QL baja de precio

A comienzos del verano, todo auguraba problemas para **Sinclair:** los grandes *stocks* sin vender repercutieron en una falta de liquidez de la empre-



sa,que no podía hacer frente a sus deudas. La solución milagrosa, llamada **Maxwell**, consistía en la capitalización por parte de este editor inglés, a cambio de la cesión del control de la compañía. Este compromiso dejó respirar a sir **Clive** hasta el verano.

Pero a primeros de agosto **Sinclair** anuncia que ya no necesita el acuerdo con **Maxwell**, y se rompen las negociaciones. El tío **Clive** ha firmado un acuerdo con una cadena de tiendas de electrónica, por el que éstas adelantan dinero a cambio de descuentos sobre las ventas. El siguiente golpe de su agresiva estrategia fue bajar a la mitad el precio del QL: de 400 a 200 libras, con fecha de dos de septiembre. **Investrónica** no ha tardado en reaccionar, y ha puesto el precio de la máquina española, que costaba 125.000 pesetas, en 79.000 pesetas. Buenas noticias para los que todavía no han pasado por la tienda.

## Un disco para ahorrar infartos

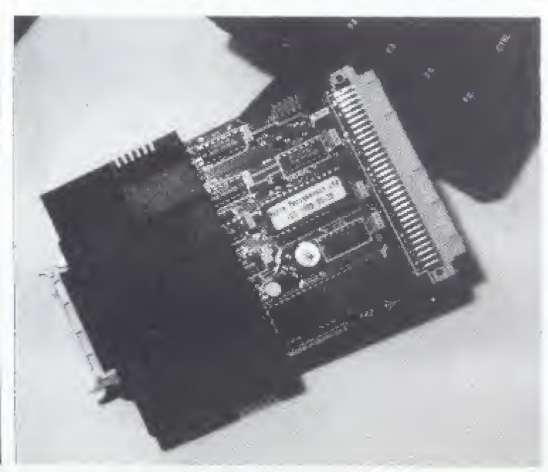
El principal problema del QL a estas alturas es la dependencia de los *microdrives*. Si bien su calidad ha mejorado, siguen ofreciendo una capacidad baja, y no toleran bien el esfuerzo que supone utilizarlos en aplicaciones con gran movimiento de datos. Además, su fiabilidad no llega al *standard* profesional. Por esta razón, **Investrónica** comercializará la unidad de disco «oficial» **Sinclair**. Se trata de un disco de 3"1/2 de doble cara, con una capacidad de 720K formateadas. Los programas de **Psion** versión española trabajan perfectamente con la nueva unidad tras una pequeña adaptación.

La unidad incluye en su ROM una ampliación del BASIC que proporciona comandos de acceso aleatorio y otras ayudas de interés. Importante para un uso profesional, por su capacidad, rapidez y fiabilidad. El precio ha sido fijado en 93.000 pesetas con controlador. Aunque resulte algo caro, más de un usuario respirará aliviado.

## Más aplicaciones: Nóminas

Cuando, el mes pasado, pasamos revista a los programas de aplicación de origen español que iban apareciendo, no nos había llegado todavía el programa de nóminas de **WorldMicro.** Se trata de una aplicación escrita sobre ARCHIVE, con cuatro bases de datos relacionadas: contienen el Convenio colectivo, los datos de la Seguridad Social (Bases y Tipos) y el fichero de empleados, respectivamente. El programa permite actualizar cualquiera de las cuatro y listar la nómina, los recibos y los datos de la Seguridad Social.

Un programa sencillo para una tarea muy mecanizable. En el listado de recibos está prevista la inclusión de incidencias. La salida está adaptada a los impresos oficiales, requiriendo una impresora *standard* de 80 columnas.





# Jugar con el QL

Los primeros compradores del QL disimulaban su disgusto por la falta de software para jugar diciendo, que «al fin y al cabo se trata de una máquina seria: dispone de más compiladores y aplicaciones profesionales que ningún ordenador de su precio...». Lo cierto es que los buenos programas de juego tardaron, pero van haciendo su aparición.

Se dispone ahora de uno de los mejores programas de ajedrez para microordenadores (ya comentado en un número anterior) y de un Backgammon bastante interesante. Está anunciada, aunque todavía no la hemos podido ver en España, la versión QL del Match Point, el mejor juego de simulación tenística. Las fotos de la pantalla dejan ver unos gráficos excelentes; esperamos que el movimiento sea tan bueno como en el Spectrum.

D Day, un juego de guerra ya famoso, también aparece en una generosa versión, con dos microdrives que forman uno de los juegos de guerra más completos que hemos visto. Por otra parte, han ido apareciendo una serie de juegos de acción de dudosa calidad, sobre todo teniendo en cuenta las grandes posibilidades de la máquina.




Entre ellos podemos citar Star Guard & Galactic Invaders, un microdrive con dos juegos, de los que el primero es realmente malo, con un movimiento muy difícil y una dinámica que, aunque a priori atractiva, queda oculta tras la dificultad para mover nuestro láser. Galactic Invaders es una versión (¡otra!) del viejo Space Invaders; el juego es simpático, y la idea, una de las más explotadas de la historia del software, sigue siendo atractiva a pesar del tiempo transcurrido.




Night Nurse es otro juego que no pasa el listón mínimo de calidad: se trata de acompañar las andanzas de una Enfermera en su guardia, llegando hasta una fiesta que se celebra en una de las salas del hospital. Lento y poco elaborado, lo único bien resuelto es el movimiento de los personajes.

Comentamos a continuación dos de estos programas: Super Backgammon, y D Day. Ambos nos han parecido lo mejor que ha llegado a nuestras manos hasta el momento. Afortunadamente, parece que el déficit de juegos se va llenando poco a poco, y que en futuros números podremos ampliar este breve catálogo.

# QL D-Day







Hay dos tipos de usuarios de ordenadores: los que disfrutan con los Juegos de Guerra y los que no. Para los fanáticos de la estrategia, debe ser bienvenido el lanzamiento de D-Day en su versión QL. Ha sido éste uno de los primeros programas que han llegado a llenar la carencia de programas de juego para esta máquina.

Producido por Games Workshop y distribuido en España por Serma, el Día D participa de todas las características de los juegos de guerra clásicos, y la mayor memoria del QL ha permitido incorporarle una gran variedad de nuevas opciones. Los fanáticos de la estrategia y la táctica disfrutarán, sin duda, con este programa, ya que se puede jugar contra un contrincante humano, usando el ordenador para encargarse del trabajo pesado, y así poder concentrarnos en el juego, o bien luchar contra la máquina, con opciones algo más limitadas.

Se trata de una simulación del combate táctico en la Segunda Guerra Mundial, en los momentos que siguieron al desembarco de Normandía. Disponemos de cuatro escenarios, con objetivos distintos para cada ejército, y se pueden seleccionar con gran libertad las condiciones iniciales de la batalla.

En caso de combatir dos jugadores, pueden elegir entre «comprar» sus ejércitos o utilizar las opciones por defecto del programa que incluyen selección entre ejércitos de 15 a 50 unidades. También podemos, en este caso, colocar nuestras unidades en la disposición que deseemos. Si jugamos contra el ordenador, en cambio, sólo podremos elegir entre dos ejércitos de 15 ó 50 unidades y jugar en el bando aliado o alemán.

El tiempo se suele convertir en un condicionamiento importante en los juegos de guerra, ya que no resulta fácil acabar la partida, incluso con el ejército más pequeño, en manos de un par de horas. Para no dejar las partidas a medias se nos da la opción de salvar el juego en microdrive; al volver a arrancar el programa continuaremos automáticamente donde lo dejamos.

Como se nos anuncia en la carátula, el juego contiene casi 200K de programa y datos, con alrededor de 20K dedicadas a «inteligencia artificial». Con esos tamaños, los tiempos de carga son grandes, ya que el programa lee gran cantidad de datos de los dos cartuchos. El primero contiene los programas, mientras que en el segundo se almacenan los mapas y características de las unidades.

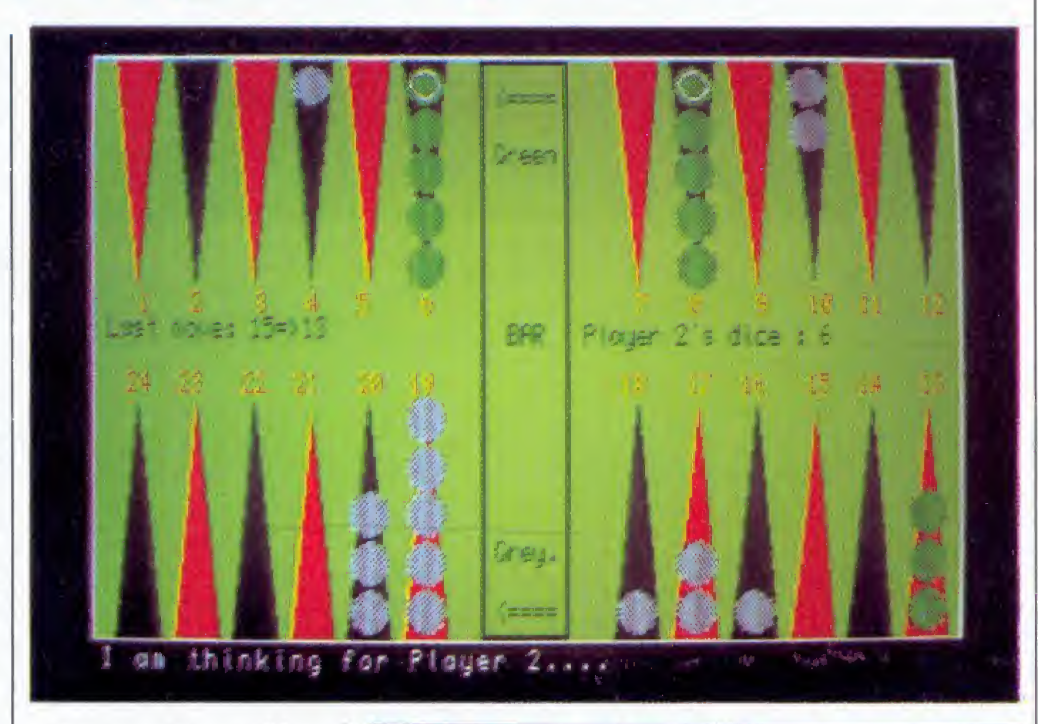
Durante el período de carga se puede elegir entre uno o dos jugadores, así como el tipo de ejército que se va a utilizar. En la selección de unidades el programa resulta una vez más muy completo, con más de diez tipos en cada bando, incluyendo barcos de transporte y camiones, que pueden mover con rapidez nuestros efectivos. Existen diferentes unidades de carros de combate, que difieren en capacidad ofensiva, precio y prestaciones. Para seleccionar nuestra fuerza de invasión, el programa nos da dinero en proporción al tamaño del cuerpo de ejército. Si lo gastamos sin cumplir nuestro número de unidades, el ordenador nos proporciona unidades de infantería ligera hasta completar el número preestablecido. Así podemos elegir entre un ejército equilibrado o bien otro con unas pocas unidades selectas y un gran número de batallones de infantería ligera, de pocas posibilidades en combate.

Es en la dinámica del juego donde se le pueden poner las mayores objeciones: el movimiento de las unidades se realiza situando el cursor sobre cada una, eligiendo la modalidad (manual o automático), y eligiendo la dirección de cada paso (modo manual) o del movimiento entero (automático). El principal problema es que el posicionamiento del cursor resulta algo lento, y no disponemos de ningún sistema para posicionarnos rá-

pidamente sobre el mapa.

Con una presentación satisfactoria, unos gráficos muy correctos si disponemos de TV color (en blanco y negro resultan algo más confusos, aunque se distinguen los distintos tipos de terreno sin dificultad), y, como punto más débil, un movimiento algo lento por el mapa. Creemos que hará las delicias de los aficionados a estos juegos e incluso puede llamar la atención a las personas que nunca han jugado a la guerra. En cualquier caso, conviene no tomárselo demasiado en serio.

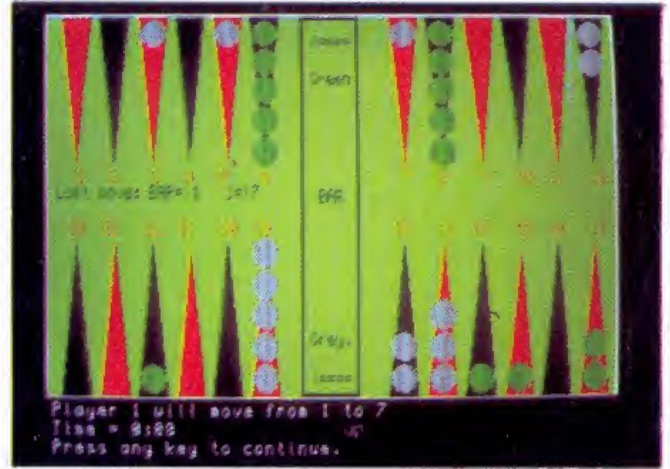
# QL Super Backgammon




A pesar de que los investigadores en Inteligencia Artificial concentraron sus esfuerzos en juegos inteligentes como el ajedrez, pronto se dieron cuenta de que resultaba muy difícil introducir elementos de análisis de posición con cierto éxito. Hace alrededor de cinco años se eligió el Backgammon como juego ideal, por la simplicidad de sus reglas, para lograr un programa «experto», guiado por ideas generales, que permitiera un juego realmente bueno. Este programa derrotó, con sorprendente facilidad, al Campeón del Mundo de la especialidad, siendo el primer juego en el que se lograba algo parecido.

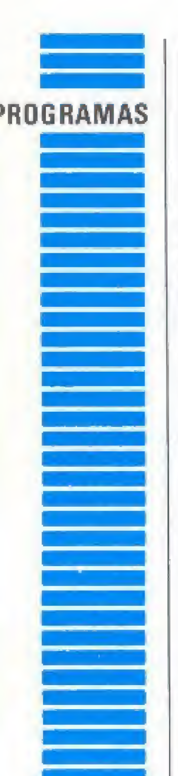
El programa que nos ofrece Digital Precision, y en España Serma, no llega a tanto: un jugador medio le vence con cierta facilidad en el nivel más alto, aunque es una buena herramienta para aprender las sutilezas de este juego, mucho más difícil de lo que parece. El juego consiste en cruzar, cada jugador en dirección contraria, el tablero. Se puede comer las piezas que queden aisladas, y cada jugador puede sacar sus piezas del tablero cuando consigue introducirlas todas en su cuadrante. Gana el primero que saque todas sus piezas.

El programa ofrece una novedad curiosa: incluye un documento QUILL donde se explican con detalle las reglas del juego. Una vez arrancado debemos decidir si el ordenador juega contra sí mismo, si juegan dos personas o si hemos decidido enfrentarnos contra él. En este caso podemos elegir el nivel, desde uno a seis.

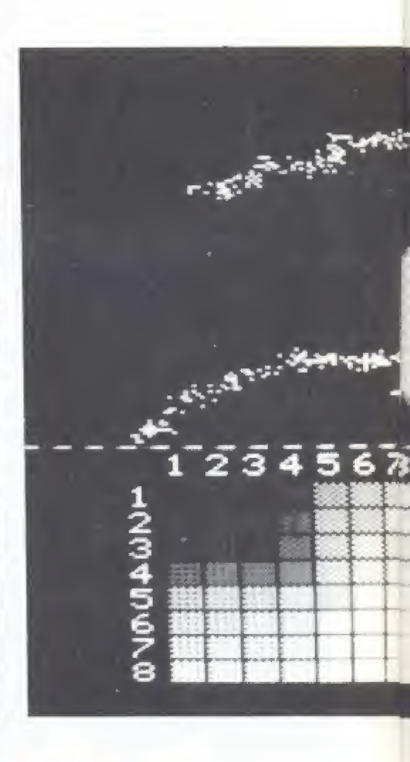



Se nos permite tirar los dados a nosotros o bien fiarnos de las tiradas del ordenador. Por otra parte, existe una opción para que el programa enseñe las jugadas que está considerando en cada momento.

En cuanto a la calidad de su juego, el programa se porta bastante bien, aunque se le puede derrotar con facilidad cuando conocemos sus debilidades. En determinadas situaciones su juego es demasiado conservador, por lo que un planteamiento ambicioso acabará con él sin problemas.






# Graphiql. Para dibujo artístico

El QL es una máquina con grandes posibilidades gráficas. EASEL, el programa de dibujo de Psion, es una buena prueba de ello. También el SuperBasic dispone de unas excelentes instrucciones gráficas. Pero no había aparecido hasta ahora ningún programa que explotase las posibilidades «artísticas» del ordenador. Graphiql, de Talent y distribuido por Serma, es el primer programa de dibujo que ha llegado a nuestras manos.

Graphiql trabaja en el modo de baja resolución de la máquina (256 x 256 puntos, con ocho colores), y resulta muy potente aunque, como veremos, también es difícil de dominar.

Tras la carga del programa nos encontramos con una pantalla negra, donde aparece una pequeña cruz de color blanco: el cursor. En el uso más sencillo, podemos usar un joystick (o las teclas del cursor) para mover el cursor. Si usamos simultáneamente la barra espaciadora (botón de disparo), el cursor irá dejando un trazo del color activo. Si introducimos PT, los movimientos del cursor dejarán la traza aunque no pulsemos la espaciadora. Para cambiar el color, basta pulsar la tecla correspondiente a éste (0-7). El comando más útil para el principiante es H, que nos presenta unas cuantas páginas de ayuda, en las que aparece una breve descripción de cada comando.

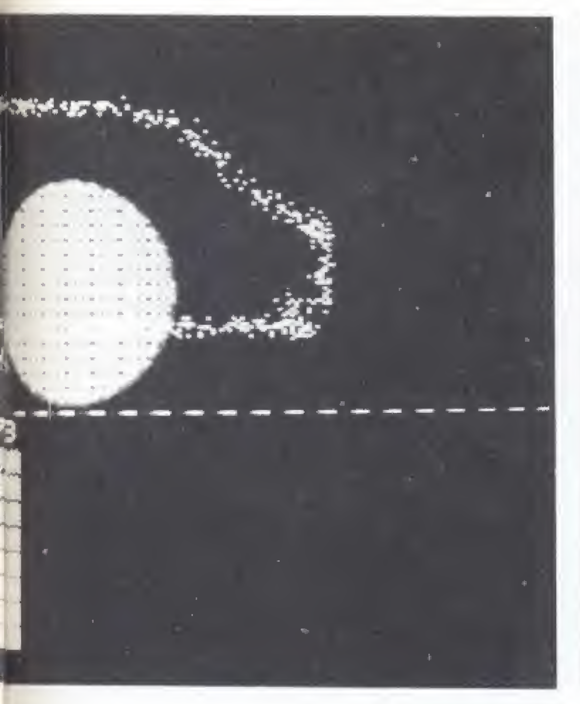
Dos modos de dibujo en la pantalla, el normal y el XOR, en el que el color que aparece en la pantalla es función de la tinta actual y el color previo en la pantalla, nos permiten obtener curiosos efectos. Asimismo, el programa dispone de comandos para dibujar líneas, círculos, elipses y rectángulos. En todos los casos, el programa nos deja ver las características del dibujo según se realiza, pudiendo abortar el comando si el resultado no es satisfactorio.

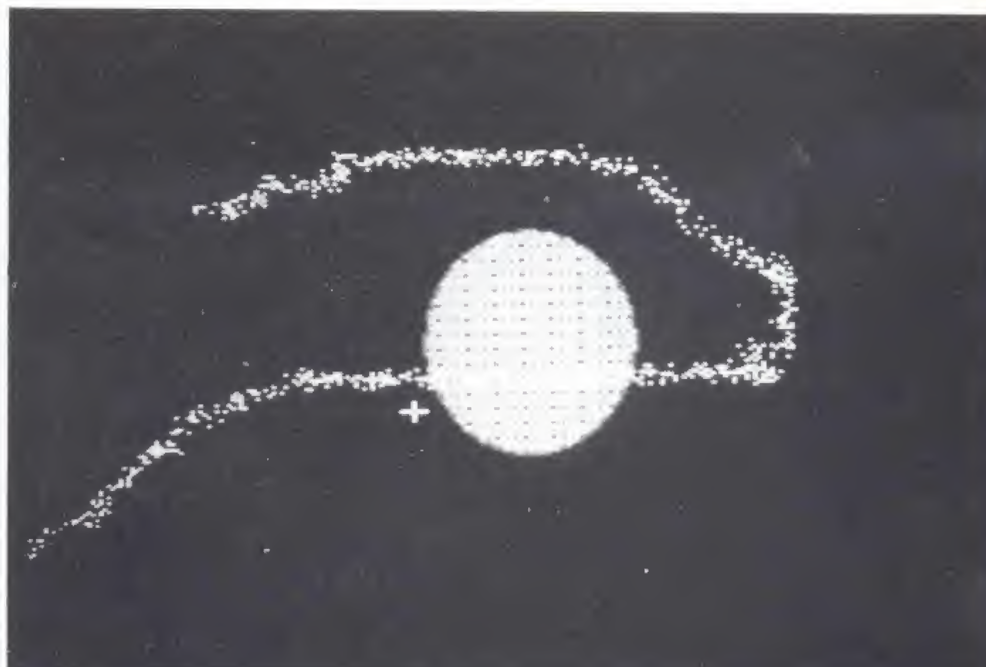
Para posicionarnos en la pantalla con más precisión, el programa dispone de un índice de la posición actual, el color seleccionado y otras características. Este índice se puede eliminar de la pantalla o cambiar de posición. Otra opción interesante es la de zoom, que permite 16 aumentos para afinar nuestra colocación en la pantalla.

Hasta aquí el programa, aunque potente, no se diferencia en nada de programas de dibujo convencionales para otras máquinas. Pero los comandos de bloques y texturas y otras opciones multiplican su potencia.

Las texturas constituyen ya un importante avance sobre los programas convencionales: se nos permite definir hasta 26 zonas rectangulares en una paleta auxiliar que aparece en la parte inferior de la pantalla y memorizarlas, para luego repetirlas en nuestros dibujos.

Casi todos los programas de dibujo disponen de una opción para rellenar áreas de un color. Graphiql va mucho más allá: el comando FA rellena de blanco una zona de la pantalla y la memori-





```
Colours are selected by the digits
0-7.
        0        Black      1
        2                   3      Purple
        4        Green      5      Cyan
        6        Yellow     7      White

  Cdigit   Change the cursor colour

The colours of pixels in the current
block can be changed by the commands:

        CS       Colour Swap
        CM*      Colour Merge

Both are followed by a pair of digits
These give the colours to be swapped
or merged.

MORE
```

za. Una vez acabado el proceso, los comandos de rellenado se referirán a ese área. Si, por ejemplo, tecleamos FC, la zona aparece dibujada en el color activo. Eligiendo otro color y repitiendo el comando, se obtiene un cambio de la zona al nuevo color. Por último, si introducimos FT, el programa volverá a rellenar la zona, esta vez con la textura activa. Así se pueden sombrear o «empapelar» nuestros dibujos sin perder gran cantidad de tiempo. Las texturas disponen de otra opción muy interesante: se le puede indicar al programa que utilice sólo parte de los colores que la componen. Así, una figura sobre fondo negro, si le decimos al programa que no use el color negro, aparecerá sobre el fondo del dibujo sin alterar el resto de éste.

Otra opción interesante es la posibilidad de usar «aerógrafo» en nuestros dibujos. Esta opción consiste en que, cada vez que cambiamos de posición el cursor, el programa dibuja una serie de puntos aleatorios del color activo alrededor. Este efecto simula un aerógrafo al que se le estuviera acabando la pintura y elimina la precisión de trazado.

Un programa que supera en mucho las posibilidades de las utilidades de dibujo de otros ordenadores, con funcionamiento rápido de sus opciones. El principal inconveniente es el modo de comandos, ya que la pantalla no hace eco de las letras introducidas, lo que multiplica las posibilidades de error, y hace que cueste trabajo familiarizarse con el programa. Una vez se profundiza en su uso, Graphiql resulta una herramienta realmente profesional, que exprime las posibilidades gráficas de la máquina. Si el proceso de selección de comandos se realizase mediante un menú gráfico, el programa merecería un 10. En su forma actual, se queda con un notable.

# CUB Y QL LA PAREJA PERFECTA

Este monitor en color CUB es TOTALMENTE compatible con el Sinclair QL.
Combina las capacidades de gráficos destacados con la facilidad de visualizar textos en 85 columnas mientras hace justicia al potencial de color del QL. Su nuevo diseño dotado de una base basculante y giratoria armoniza perfectamente con la simple apariencia funcional del QL.
Lo mejor de todo, es su precio, solo de

## 91.600

P.V.P.

### ESPECIFICACIONES

MODELO-CUB 1451/DQT3
MONITOR DE 14"
ENTRADA - RGB/TTL
RESOLUCION - 653 (H) × 585 (V)
DOT PICH - 0,43 mm.
ANCHO DE BANDA - 18 MHz
ANTIREFLEJO CRT



CUB 1451/DQ



CUB 1451/DQT 3
CON BASE BASCULANTE Y GIRATORIA






**multilogic**

Pº de la Habana, 145
Telf. 458 74 75 - 28036 Madrid

# Un nuevo operativo para el Spectrum

②

***Este mes comenzamos a daros el listado ensamblador del programa, ampliamente comentado, con lo que los que lo deseéis podréis modificarlo introduciendo nuevos comandos o cambiando los ya existentes, tanto los nuevos del programa como los antiguos del BASIC.***

Antes de comenzar hay que hacer una advertencia a los que hayan preferido esperar para teclear el programa desensamblado en lugar de las largas líneas de DATAs del mes pasado. El programa es muy largo y en algunos ensambladores puede presentar problemas para manejar en memoria todo el código fuente. Por ejemplo, si usáis el GENS3M2 de HISOFT tendréis que recurrir a ensamblar parte del código desde microdrive o cinta. En el de ARTIC si cabe todo el código (dejando aparte las interrupciones, que son un módulo totalmente independiente), pero tiene el inconveniente de que el código objeto debe ocupar las mismas direcciones que el ensamblador, y éste se destruye. Si usáis este ensamblador tendréis que ensamblar en otra dirección y luego ingeniároslas para reubicarlo. De todas formas os damos también un listado de la tabla de etiquetas para que cada uno pueda trocear el programa como más le guste y poder así ensamblar a cachos.



*Manuel Arana (a la derecha) con Pedro Verduras. Ambos son estudiantes de la Escuela Técnica Superior de Ingenieros Industriales de Madrid y comparten su afición por el Spectrum.*

## La idea

La idea de escribir este programa surgió ojeando el libro «The complete spectrum ROM disassembly», y pensando en las importantes mejoras que podrían introducirse si pudiéramos modificar un poco el programa de la ROM. Pero esto no era posible, y la única alternativa era copiar las partes

```
        ORG  *EAFF
        DEFW INT            El vector de interrupcion apunta
                            al comienzo de la rutina.
INT     PUSH AF             Guarda los registros en la  pila
        PUSH HL             durante la interrupcion.
        LD   HL,(*5C78)     Incrementa el  contenido  de  la
        INC  HL             variable del sistema FRAMES,
        LD   (*5C78),HL
        LD   A,H
        OR   L
        JR   NZ,KEYINT      y el tercer byte si es necesario
        INC  (IY+64)
KEYINT  PUSH BC             Guarda los demas registros.
        PUSH DE
```




fundamentales en RAM y realizar ahí los cambios. Para empezar es necesario hacernos con el control del bucle principal que se encarga de llamar a las rutinas que realizan las funciones más importantes, como podrían ser el pedirnos la introducción de una línea BASIC, analizarla o ejecutarla. A partir de ahí podríamos ir sustituyendo las rutinas necesarias para la correcta interpretación de los nuevos comandos. Además para que éstos puedan ser tecleados directamente hay que hacerse con el control de las interrupciones. Y ya puestos a construir un nuevo sistema, ¿por qué no dotarle de un modo de presentación de 64 caracteres? Aparentemente no presenta muchos problemas, basta con hacer que el canal «S» apunte a la dirección donde se encuentra la rutina de presentación cuando se seleccione este modo. Sin embargo, nos encontramos con que en bastantes puntos se vuelve a colocar como dirección de salida la normal de la ROM. Por esta razón tuvimos que arreglar muchas rutinas relacionadas con la escritura, como la que produce los listados automáticos o borra la pantalla. Esta es una de las causas que hicieron que el programa sea mucho más largo de lo pensado en un principio.

Para que los nuevos comandos sean interpretados correctamente, lo primero que necesitamos es asignarles un código por el que puedan ser identificados dentro de un programa BASIC. En el libro de instrucciones encontramos una sugerente oferta. Los códigos 24 a 31 no están utilizados. Aquí hay sitio para introducir hasta ocho nuevas palabras clave. Sin embargo, luego descubrimos que estos códigos en realidad son utilizados internamente por la rutina que lee el teclado para indicar que ha sido pulsada una combinación de teclas correspondiente a cambio de tinta. Esto tuvo que ser modificado para hacerlo compatible con el nuevo sistema. Por esta razón cuando está conectado el Interface 1 (las interrupciones se tratan normalmente) las palabras clave aparecen en lugar de los códigos de cambio de tinta. Los códigos al final han quedado de la siguiente forma:

24.– ELSE
25.– ON ERR
26.– AUTO
27.– RENUM
28.– ON
29.– SWAP
30.– MODE
31.–

Este último no se utiliza para nada y simplemente está como un posible octavo comando. De todas

```
        CALL KBOARD        Lee el teclado.
        JP   *004D         Vuelve a la ROM donde recupera
                           los registros y acaba la rutina.
KBOARD  CALL *028E         Llama a KEY-SCAN.
        RET  NZ            Acaba si no hay ninguna tecla
                           pulsada o hay mas de dos.
        LD   HL,*5C00      Primer grupo de variables.
KSLOOP  BIT  7,(HL)        Comprueba que esta libre.
        JR   NZ,KCHSET
        INC  HL
        DEC  (HL)          Decrementa el contador de numero
        DEC  HL            de pasadas.
        JP   NZ,KCHSET     Deja el grupo libre si ya se han
        LD   (HL),*FF      cumplido las 5 pasadas.
```

formas el número de éstos no está limitado a 8, ya que tenemos los códigos 0 a 5 en las mismas condiciones, y además en caso de necesitar todavía más códigos podríamos hacer que uno de ellos sirviera de prefijo para los demás. El límite está en vuestra imaginación.

Como, ya hemos dicho antes, para escribir el programa utilizamos el libro del listado de la ROM, y he procurado respetar en el listado los nombres de las rutinas que aparecen en él. Quizás pecando en ocasiones de abreviar las abreviaturas.

## Interrupciones

El tratamiento de las interrupciones está en primer lugar en el programa por la necesidad de colocar el vector de interrupción en una dirección concreta. Cuando el Z-80 se encuentra en el modo de interrupción 2 la dirección de la rutina se forma por el registro I como parte alta y por el dato suministrado por el dispositivo interruptor como parte baja. En el caso del Spectrum este dato es siempre FF. En nuestro programa el vector queda en la dirección EAFF y apunta a la rutina que está justo a continuación.

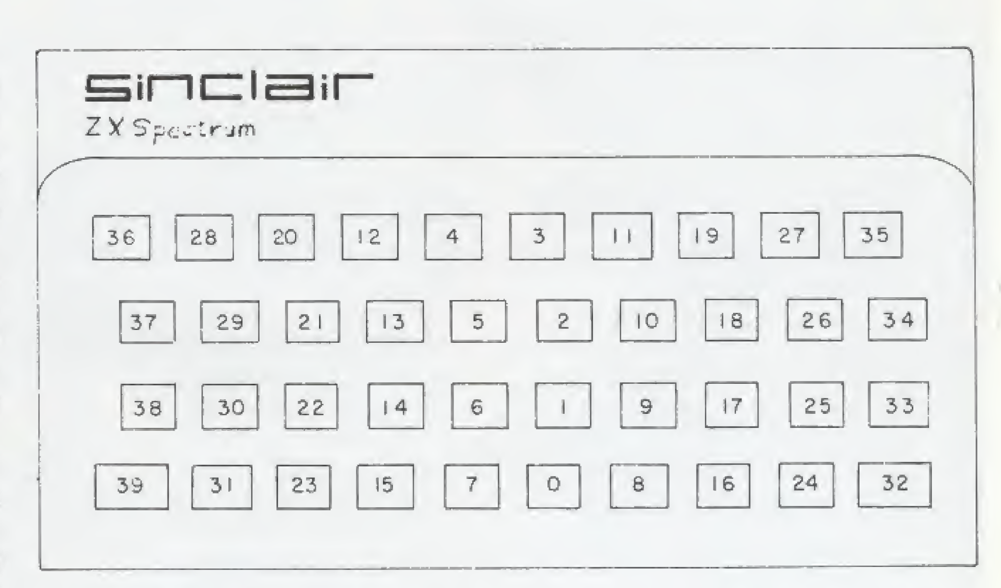
En cuanto a la rutina en sí es prácticamente igual a la de la ROM y sólo se diferencia en dos detalles, la forma de pasar los códigos de cambio de tinta y el desdoblamiento del modo extendido.

Con el Spectrum funcionando normalmente, si se pulsa en modo extendido la tecla CAPS-SHIFT y un número entre 0 y 7, la rutina K-DECODE devuelve como código de la tecla pulsada un valor entre 24 y 31 para indicar que se trata de cambio de tinta (24 para INK 0 y 31 para INK 7). De la misma forma utiliza los códigos 16 a 23 para indicar cambio de papel, pero éstos no nos importan. La solución adoptada en el programa para que estos datos no sean confundidos con los códigos de las palabras clave es devolver el dato 16 para los cambios de tinta y el 17 para los de papel, e indicar el color seleccionado dejando su número en la variable del sistema KDATA. Aquí había un pequeño error en el listado de DATAs y cuando queríamos cambiar de tinta cambiaba el papel y al revés. Si queréis solucionarlo basta con hacer POKE 60355,192.

La otra diferencia está en la misma asignación del teclado. Había que hacerla de forma que no desapareciese ninguna de las funciones ya existentes. Para ello recurrimos a desdoblar el modo extendido. Cuando se quiere acceder, por ejemplo, a la instrucción VERIFY

```
KCHSET LD   A,L
       LD   HL,^5C04       Repite la operacion con el otro
       CP   L              conjunto de variables.
       JR   NZ,KSLOOP
       CALL ^031E          K-TEST se encarga de obtener el
                           codigo de la tecla pulsada.
       RET  NC             Vuelve si la combinacion de
                           teclas no es valida.
       LD   HL,^5C00       Comprueba si la tecla ya estaba
       CP   (HL)           antes en el primer grupo,
       JP   Z,^0310
       EX   DE,HL
       LD   HL,^5C04       o en el segundo, y si es asi
       CP   (HL)           salta a la ROM donde considera
       JP   Z,^0310        la repeticion.
       BIT  7,(HL)         Mira a ver si alguno de los dos
       JP   NZ,KNEW        grupos de variables esta libre
       EX   DE,HL          para una nueva tecla.
       BIT  7,(HL)
       RET  Z
KNEW   LD   E,A            Coloca la nueva tecla en el
       LD   (HL),A         conjunto que este libre
       INC  HL
       LD   (HL),^05       contador de 5 pasadas
       INC  HL
       LD   A,(^5C09)      tiempo de repeticion
```

hay que hacerlo en modo extendido y pulsando simultáneamente la tecla R y cualquiera de los dos SHIFTs. La idea básica consiste en hacer que esto sólo se pueda hacer con uno de ellos (SS) y dejar el otro (CS) para introducir las nuevas palabras clave. De esta manera podemos introducir de golpe 26 asignaciones nuevas sobre el teclado sin que desaparezca ninguna de las antiguas. La mayor parte de ellas se han realizado de forma que coincidan con las antiguas y sólo se han cambiado las correspondientes a nuestros comandos y a GOTO y GOSUB que presentaban algunos problemas a la hora de teclear las líneas ON-GOTO. La distribución en el teclado se ha hecho sin seguir ningún criterio especial y por si a alguien no le gusta daremos un método para redefinirla. Los que tecleen el listado ensamblador basta con que modifiquen las líneas 76 a 84. Para los que teclearon el listado del mes pasado, deben cargar el programa y si por ejemplo quieren asignar la instrucción ELSE a la tecla E deben hacer POKE 60233+CODE "E", 24. Se procedería de la misma forma para cualquier otra tecla o cualquier otra cosa que quisiéramos asignar. Cuando acabéis no tenéis más que volver a grabar el programa y ya tendréis vuestra propia versión del programa con vuestras asignaciones.




```
50          LD    (HL),A
51          INC   HL
52          LD    C,(IY+7)       MODE
53          LD    D,(IY+1)       FLAGS
54          PUSH  HL
55          CALL  DECODE         Decodifica la tecla y
56          JP    ^0306          acaba en la ROM.
57 DECODE   LD    A,E
58          CP    ^3A            Salta si se trata de  un  numero
59          JP    C,KDIGIT       ,SPACE,ENTER o algun SHIFT.
60          DEC   C              Decrementa MODE
61          JP    M,^034F        salta si es modo K,L o C
62          JR    Z,KELET        Modo E
63          ADD   A,^4F          Graphics
64          RET
65 KELET    LD    HL,^01EB       Base para modo E sin SHIFT
66          INC   B              Hay algun SHIFT ?
67          JR    Z,LOOKUP       salta si no
68          LD    HL,^0205       Base para modo E + SS
69          PUSH  AF
70          LD    A,B
71          CP    ^19            Se trata de CAPS SHIFT ?
72          JR    Z,NOCS         Salta si no
73          LD    HL,TABC2-^41   Base para modo E + CS
74 NOCS     POP   AF
75 LOOKUP   JP    ^034A          Acaba en la ROM
```



N.º 4
PRIMERA REVISTA EN CASSETTE
MAYO 85
SPECTRUM COMPUTING
PARA 16K Y 48K
3 D
Añada una nueva dimensión a su SPECTRUM. Acción en tres dimensiones. Busque y destruya la flota de ataque estelar. Piérdase en nuestro laberinto en tres dimensiones en código máquina.
Defensa
Su oportunidad para venir en defensa del planeta.
875 Ptas.
Más sobre la programación de dibujos animados.
Clocks up
¿Sabe generar un reloj digital en su SPECTRUM?
Y MUCHO MAS
ZX Spectrum
BIENVENIDO A
SPECTRUM COMPUTING
LABERINTO EN TRES DIMENSIONES
GRAFICOS
WIZARD
RELOJ
DEFENSA
DIBUJOS ANIMADOS
ATAQUE ESTELAR
AGENDA TELEFONICA
SKI
PUZLE
LA SERPIENTE
MAS DE 150.000 PTAS. EN PREMIOS. BASES EN EL INTERIOR
CUPON DE PEDIDO
Nº 4

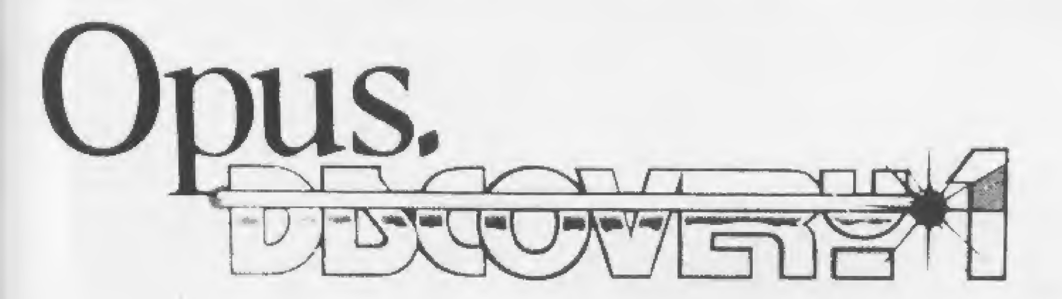
Todo esto solamente es válido para los que no utilicen el Interface 1, ya que cuando éste está conectado no se utiliza la rutina de interrupción. De hecho aquí está una de las causas de incompatibilidad con el interface. En el tratamiento de las interrupciones se utilizan varias rutinas de la ROM que cuando ésta es paginada por el interface no se encuentran en su sitio, lo que provoca que la máquina se «cuelge» al intentar llamarlas. Teóricamente sería posible que la rutina de interrupción antes de hacer nada comprobase cuál de las dos ROMs está en ese momento y procediese en consecuencia. Sin embargo, nuestro único intento en ese sentido no ha tenido muy buenos resultados, ya que el Spectrum se colgaba en algunas ocasiones al acabar de ejecutar algún comando del interface. Nosotros seguimos pensando que es posible y quizás alguno de vosotros encuentre la solución.

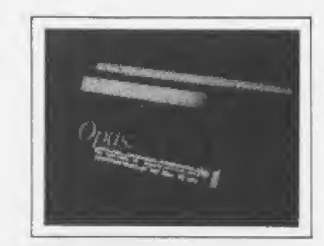



Por lo demás las interrupciones se tratan normalmente y sólo queda destacar por su utilidad dos rutinas de la ROM. La primera es KSCAN en la dirección *028E, que es la que realmente lee el teclado. Si sólo hay pulsada una tecla nos devuelve en el registro E un número entre 0 y 39 distinto para cada una de las 40 teclas tal como indica la figura 1. Si son dos las teclas pulsadas nos devuelve el número de tecla más bajo en el registro D, y el otro en el E, a no ser que

```
                                     Tabla modo extendido + CS
 76 TABC2  DEFB "~^",^DC,^DA          A  B  C  ~.       BRIGHT  PAPER
 77        DEFB ^5C,^1A,^7B           D  E  F  barra    AUTO    llave
 78        DEFB ^EC,^ED,^BF           G  H  I  GOTO     GOSUB   IN
 79        DEFB ^AE,^AA,^AB           J  K  L  VAL$     SCREEN$ ATTR
 80        DEFB ^DD,^DE,^DF           M  N  O  INVERSE  OVER    OUT
 81        DEFB ^7F,^18,^1B           P  Q  R  copyr.   ELSE    RENUM
 82        DEFB ^7C,^1C,^1D           S  T  U  barra    ON      SWAP
 83        DEFB ^DB,^19,^D9           V  W  X  FLASH    ONERR   INK
 84        DEFB ^1E,^D7               Y  Z     MODE     BEEP

 85 KDIGIT CP   ^30              Acaba si se trata de SPACE,
 86        RET  C                ENTER o algun SHIFT.
 87        DEC  C
 88        JP   M,^039D          Salta con los modos K,L y C
 89        JP   NZ,^0389         tambien con graphics
 90        LD   HL,^0254         Base para modo E + SS
 91        BIT  5,B
 92        JP   Z,^034A          Salta si esta pulsada SS.
 93        CP   ^38              Comprueba si se trata de los
 94        JP   NC,^0382         numeros 8 o 9
 95        AND  ^07              Se queda con el numero del color
 96        LD   (^5C0D),A        y lo deja en K DATA.
 97        LD   A,^10            Devuelve el codigo de cambio de
 98        INC  B                tinta,
 99        RET  NZ
100        INC  A                o cambio de papel si no esta
101        RET                   pulsado CS.
102 CHTAB  DEFS ^380             Reserva espacio para la
                                 definicion de caracteres.
103        LD   A,^EA            Pone en marcha la rutina de
```










# UN SISTEMA COMPACTO DE DISCO PARA EL SPECTRUM Y EL SPECTRUM +

El Discovery 1 es el más avanzado sistema de disco jamás desarrollado para el Spectrum.

Diseñado en Inglaterra, incorpora la más reciente tecnología japonesa en discos de 3,5''. El Discovery es el primero de la nueva generación de periféricos para el Spectrum.

## Las asombrosas características del Discovery incluyen:

- El más moderno sistema de disco de 3.5''
- Salida Centronics para impresora.
- Interface de Joystick tipo Kempston
- Salida para monitor monocromo.
- Alimentación propia y para el Spectrum
- Conector para otros periféricos.

El Discovery se conecta simplemente en la parte trasera de su Spectrum y le proporciona acceso instantáneo a la velocidad, prestaciones y seguridad de un sistema de disco.

El Discovery está especialmente diseñado para aceptar los mismos comandos que el Interface 1, permitiéndole usar la mayoría de los programas de casette diseñados para Microdrive sin necesidad de modificarlos.

El Discovery no ocupa sitio en la RAM del Spectrum por lo que podrá seguir disfrutando de sus programas de juego favoritos sin necesidad de desconectarlo ya que deja libre toda la memoria del Spectrum. Además, puede ya guardar la fuente de alimentación del Spectrum puesto que el Discovery le proporciona tensión a través del conector trasero.

El Discovery, con acceso aleatorio permite, por fin, aplicaciones serias para el Spectrum; contabilidades, facturaciones, tratamientos de textos, etc., que eran posibilidades solamente, están ahora a su alcance con el Discovery. Hay a su disposición un amplio catálogo de programas realmente útiles y otros que van apareciendo continuamente, que le permitirán aprovechar la insospechada capacidad de su Spectrum.

**Discovery 1 - El nuevo poder que respalda a su Spectrum**



## SISTEMAS INCLUIDOS

| | |
|---|---|
| **Salida para impresora** | Compatible Centronics |
| **Interface joystick** | Conector standard tipo D de nueve contactos. Compatible Kempston. |
| **Salida para monitor** | Conector standard para monitor monocromo. |
| **Conector de expansión** | De 56 vías |
| **Alimentación** | 220 V. 50 Hz. con cable e interruptor incorporado. No se precisa la alimentación del Spectrum |

## FUTURAS AMPLIACIONES

El Discovery 1 puede ampliarse a Discovery 2 en cualquier momento, colocando otra unidad de disco idéntica en el espacio previsto para ello. Todas las conexiones ya están efectuadas en el interior para simplificar el montaje.

El Discovery Plus, que es el nombre del kit de ampliación, está formado por la unidad de disco, 1 cable de 34 vías y una RAM 6116, que va insertada en un zócalo previsto a este fin.

## PROGRAMAS DISPONIBLES

- Contabilidad profesional
- Procesador de textos
- Base de datos
- etc.....



## UNIDAD DE DISCO

| | |
|---|---|
| Tipo | Standard de 3.5'' |
| Modelo | 35401 |
| Pistas | 40 |
| Caras | 1 |
| Capacidad total | 250 Kbytes |
| Capacidad formateado | 180 Kbytes |
| Tiempo de acceso pista/pista | 3 milisegundos |
| Alimentación | La del Discovery |





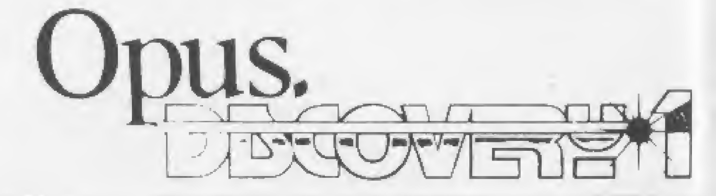








- Sistema RTTY (Rx-Tx) para Spectrum 4.800 Ptas.
- Programador de EPROMS (desde la 2716 a la 27128) *(para el Spectrum y el Amstrad 464)*
- Interface Centronics para QL 12.800 Ptas.
- Impresora Ibico 80 columnas 30.400 Ptas.




**SOLICITELO A SU DISTRIBUIDOR DE INFORMATICA O DIRECTAMENTE A**




**SISTEMAS LÓGICOS GIRONA, S.A.**
Avda. San Narciso, 24
17005 GERONA - Tel. (972) 23 71 00

una de ellas sea un shift, en cuyo caso irá siempre en el registro D. De haber más de dos teclas pulsadas nos lo indica dejando el indicador de cero a cero (NZ).

La otra rutina es KTEST, que realiza la conversión de los datos de KSCAN al código de la tecla pulsada si estuviéramos en modo C. Aquí es donde está la base de la tabla que indica qué letra corresponde a cada tecla. Si la cambiáramos conseguiríamos que nuestro teclado dejara de ser de tipo QWERTY, lo que sería un pequeño lío.

Esto es todo en lo que respecta a las interrupciones. A continuación en el programa viene la definición de los 112 caracteres del modo 1. De momento eludiremos el problema de introducirlos y por ahora limitaros a reservar 896 bytes.

## El bucle principal

El bucle principal del BASIC (MAIN) es la parte más importante del programa, ya que es el encargado de ir llamando a las demás rutinas. La figura 2 muestra el diagrama de flujo de esta parte, que como podréis ver no tiene principio ni final al estar continuamente ejecutándose.

Las líneas 103-108 son las encargadas de poner en funcionamiento el programa, preparando el modo de interrupción 2 y saltando sobre NEW. Se podría pensar que las líneas 106 y 107 no son necesarias, ya que NEW se encarga de poner en su sitio la pila. Esto es cierto, pero su finalidad es poder realizar una entrada en caliente cuando está conectado el interface 1. Esta entrada se hace después de que el programa cargador haya «pokeado» el salto sobre NEW para que entre en MAIN 1 y de esta forma no se borre el BASIC. Cuando está el interface la entrada debe hacerse sin conectar las interrupciones.

```
104           LD    I,A            interrupcion en modo 2
105           IM    2
106           LD    SP,(^5C3D)     Hace SP igual a ERR SP + 2
107           POP   AF
108           JP    NEW            Inicializa el programa.

109 AUTOFF    XOR   A              Apaga el modo AUTO. Salta aqui
110           LD    (AUTON),A      cuando pulsamos solamente ENTER.
111 MEXEC     LD    (IY+49),^02    La parte baja de la pantalla
                                   tiene dos lineas.
112           CALL  AULIST         Produce un listado automatico.
113 MAIN1     CALL  ^16B0          SET-MIN Limpia todas las
                                   areas detras de E-LINE.
114           LD    A,(AUTON)      Comprueba si estamos en modo
115           AND   A              AUTO.
116           JR    Z,MAIN2        Salta si no.
117           LD    A,^FF          Abre el canal 'R'.
118           CALL  ^1601
119           LD    HL,(AUTOLI)    Calcula el nuevo numero de linea
120           LD    DE,(AUTOIN)    sumando el incremento.
121           ADD   HL,DE
122           BIT   6,H            Comprueba si nos hemos pasado.
123           JR    NZ,MAIN2
124           LD    (AUTOLI),HL    Guarda el numero para la
125           LD    B,H            siguiente linea,
126           LD    C,L
127           CALL  ^1A1B          y 'lo 'escribe' en el canal R.
128 MAIN2     LD    A,^00          Abre el stream 0 (normalmente
129           CALL  ^1601          canal K).
130           CALL  EDITOR         Pide una linea BASIC.
131           CALL  LISCAN         Comprueba si la sintaxis es
```




El resto del bucle se diferencia del de la ROM en el tratamiento de los errores y en el modo AUTO, aparte de que llama a las rutinas propias en lugar de a las de la ROM.

El modo AUTO se controla mediante la variable del sistema AUTON, si su valor es distinto de 0 hay que colocar el número de línea. La forma de hacerlo es abriendo el canal «R» y escribiendo a través de él el número de línea. De esta forma se introduce directamente en el área de edición y cuando se llame al EDITOR aparecerá automáticamente el número.

El tratamiento de errores se realiza cuando se vuelve de ejecutar un programa BASIC. Lo primero que hace es comprobar si se trata de un error 0 (no error), en cuyo caso no debe hacer nada más. Si se trata de otro error, hay que rastrear todo el programa en busca de algún ON ERR que se refiera a él, y si se encuentra continuar la ejecución en ese punto. Esta es la causa de que con programas largos tarde un poco en responder al BREAK, ya que debe buscar por todo el programa antes de dar el mensaje de error.

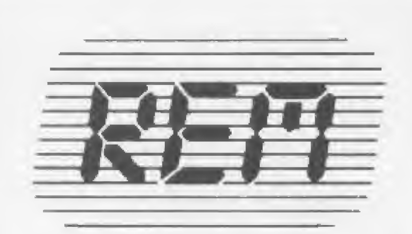
En cuanto a la asignación de la variable BASIC err se hace de la misma forma que la instrucción LET, llamando primero a CLASS-1 para inicializar la variable del sistema DEST y después a LET para que le de el valor correspondiente. Aquí se presenta un problema, que es que al poner el número de error en el STACK se modifica DEST, y ya no apunta al nombre de la variable. Por esta razón aparece dos veces en el programa.

**Manuel Arana**

```
              BIT   7,(IY+0)       correcta.
              JR    NZ,MAIN3       Salta si no hay problema.
              BIT   4,(IY+48)      Comprueba si realmente estamos
              JR    Z,MAIN4        utilizando el canal K.
              LD    HL,(^5C59)     E-LINE
              CALL  ^11A7          Elimina los numeros en coma
                                   flotante que introdujo LISCAN.
              LD    (IY+0),^FF
              JR    MAIN2          La linea debe ser revisada.
    MAIN3     LD    HL,(^5C59)     E-LINE
              LD    (^5C5D),HL     CH-ADD
              CALL  ELINO          Mira a ver si empieza con un
              LD    A,B            numero de linea.
              OR    C
              JP    NZ,MADD        Si es asi la linea debe
                                   introducirse en el programa.
              RST   ^18            GET-CHAR
              CP    ^0D            Si se trata solamente de ENTER
              JR    Z,AUTOFF       hay que salir de AUTO.
              BIT   0,(IY+48)      Borra la pantalla si es
              CALL  NZ,CLALL       necesario,
              CALL  ^0D6E          y la parte baja siempre.
              CALL  CLCHAN         Hace que los canales K y S
                                   vuelvan a apuntar a su sitio.
              LD    A,^19          Calcula el contador de scoll
```

```
        SUB   (IY+79)         S-POSN
        LD    (IY+82),A       SCR-CT
        SET   7,(IY+1)        Se va ha ejecutar una linea.
        LD    (IY+0),^FF      No hay error.
        LD    (IY+10),^01     NSPPC. Empezamos por la  primera
                              instruccion.
        PUSH  HL
        LD    HL,(^5C8A)      Guarda la variable S-POSNL.
        LD    (^5CB0),HL
        POP   HL
        CALL  LIRUN           Ejecuta la linea/programa
LOERR   LD    A,(^5C3A)       Comprueba  si  se  ha  producido
        INC   A               algun error.
        JR    Z,MAIN4         Salta si no.
        LD    C,A             Guarda el codigo del error.
        PUSH  BC
        LD    HL,(^5C53)      PROG
        LD    E,^19           Codigo de ON ERR.
        CALL  LOOKP3          Busca el primero en el programa.
        EX    DE,HL           Guarda el numero de  instruccion
        EX    (SP),HL         y el codigo de ON ERR.
        EX    DE,HL
        JR    C,NEERR         Salta si  no  se  ha  encontrado
                              ningun ON ERR.
ERRLO   PUSH  BC
        CALL  NEXTCH          Busca si hay  alguna  referencia
        CALL  LIERR           al error que se ha producido.
        POP   BC
        JP    Z,ENERR         Salta si si que la hay.
        EX    DE,HL           Recupera     el     numero     de
        EX    (SP),HL         instruccion.
        EX    DE,HL
        CALL  LOOKP2          busca el siguiente ON ERR,
        EX    DE,HL           y vuelve a guardar el numero  de
        EX    (SP),HL         instruccion  y  el  codigo    de
        EX    DE,HL           ON ERR.
        JR    NC,ERRLO        Salta si ha encontrado el ON ERR
NEERR   POP   BC              Ya no necesitamos la dirreccion.
MAIN4   HALT                  Espera la siguiente interrupcion
        RES   5,(IY+1)        FLAGS indica que se  puede  leer
                              una nueva tecla.
        BIT   1,(IY+48)       FLAGS2
        CALL  NZ,^0EDC        Copia el buffer de la impresora.
        LD    A,(IY+0)        Calcula  el  codigo  del   error
        INC   A               producido.
MAING   PUSH  AF              Lo guarda.
        LD    HL,^0000
        LD    (IY+55),H       FLAGX
        LD    (IY+38),H       X PTR hi
        LD    (^5C0B),HL      DEFADD
```

# Guía del comprador de Todospectrum




• Ordenadores personales Hard y Soft.
• Cursos de Basic.

Oficina **RENOVACION EN MARCHA, S. A.**
C/ Espronceda, 34. 28003-MADRID
Tfno. (91) 441 24 78

**REMSHOP 1**
Galileo, 4. 28015 MADRID
Tfno. (91) 445 28 08

**REMSHOP 2**
C/ Dr. Castelo, 14. 28008 MADRID
Tfno. (91) 274 98 43

**REMSHOP 3**
C/ Modesto Lafuente, 33. 28003 MADRID
Tfno. (91) 233 83 19

**REMSHOP BARCELONA**
C/Muntaner 55 - 0804 BARCELONA
Tfno (93) 253 26 18

**REMSHOP LAS PALMAS**
C/ General Mas de Gamindez, 45. LAS PALMAS
Tfno. (928) 23 02 90

**REMSHOP BILBAO**
C/ General Concha, 12 - 48008 BILBAO
Tfno. (94) 444 68 68

**REMSHOP OVIEDO**
C/ Matemático Pedrayes, 6 - 33005 OVIEDO
Tfno. (985) 25 25 95




DISTRIBUIDORES DE:
COMMODORE-64
ORIC-ATMOS
ZX SPECTRUM
SINCLAIR ZX 81
ROCKWELL'-AIM-65
DRAGON-32
NEW BRAIN
DRAGON-64
CASIO FP-200

ELECTRONICA SANDOVAL, S. A.
C/ SANDOVAL, 3, 4, 6. 28010-MADRID
Teléfonos: 445 75 58 - 445 76 00 - 445 18 70
447 42 01
C/ SANDOVAL, 4 y 6
Centralita 445 18 33 (8 lineas)




**CASSETTES DE CALIDAD PROBADA PARA ORDENADORES**

| Cada uno | Caja de 10 | Caja de 30 |
|---|---|---|
| C-5 199 ptas. | 1.393 ptas. | 3.582 ptas. |
| C-10 209 ptas. | 1.463 ptas. | 3.762 ptas. |
| C-15 219 ptas. | 1.533 ptas. | 3.942 ptas. |
| C-20 229 ptas. | 1.602 ptas. | 4.122 ptas. |

Libre de gastos de envío contra reembolso correos
CAMAFEO INC. Dep. 03
**José Lázaro Galdiano, 1. 28036 Madrid.**

## K-BITS

- **ORDENADORES PERSONALES**
- **GESTION**
- **APLICACIONES ARQUITECTURA**
- GARANTIA OFICIAL
- FACILIDADES DE PAGO
- ENVIOS A PROVINCIAS

**Barquillo 15 - Tfno.: 232 57 37 - 28004 MADRID**

CURSO DE CONTABILIDAD PARA P y M EMPRESAS

**EN ZX SPECTRUM**

- Libros Oficiales Contabilidad
- Diarios, Inventarios, Balances, etc.
- Plan General Contable

**CENTRO DE ESTUDIOS: SUMAAS**
c/. Desengaño, 12 - 3.º-3 28004 Madrid
Telfs.: 221 31 49 - 221 38 35

# ANUNCIESE por MODULOS

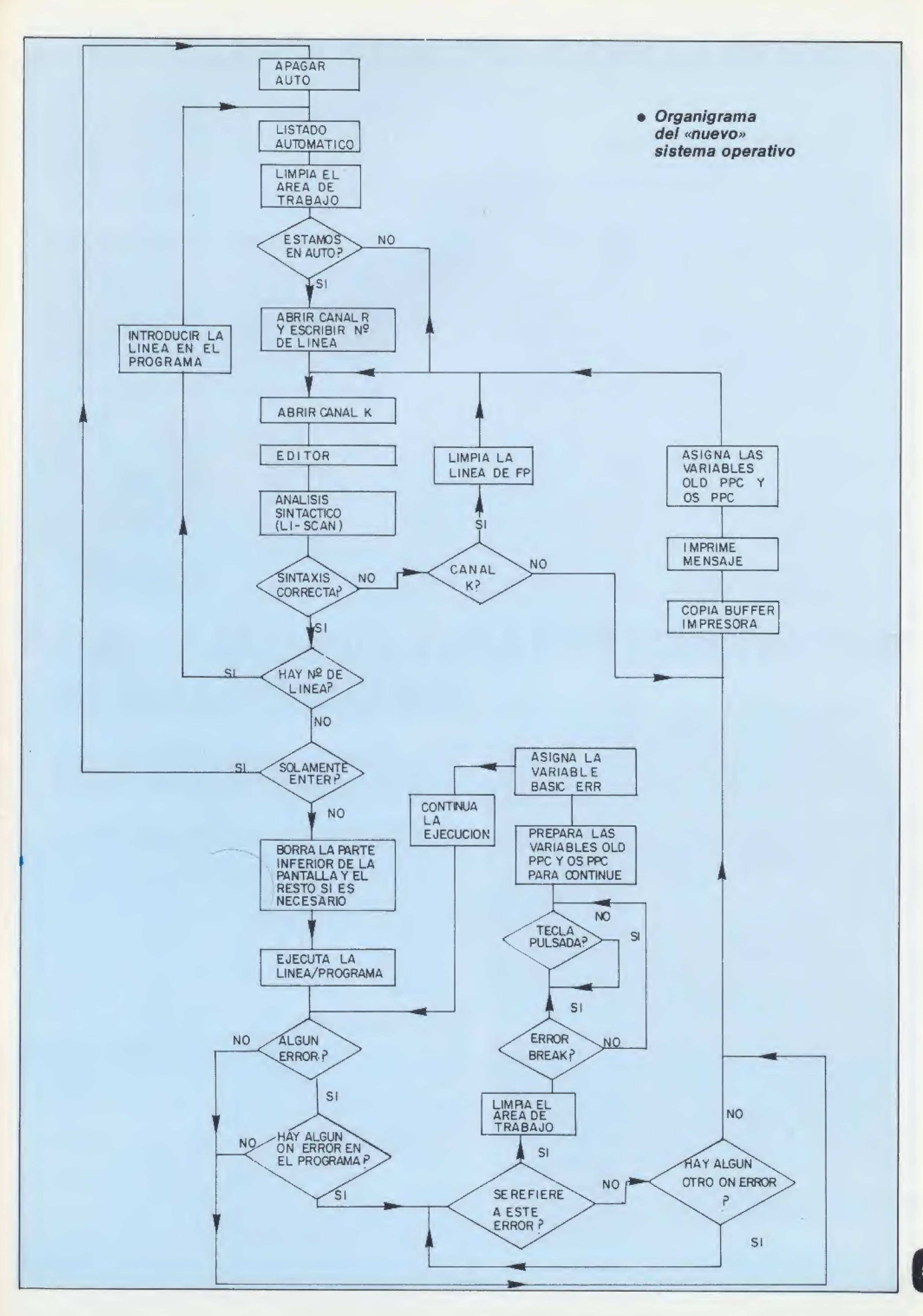
# MADRID (91) 733 96 62

# BARCELONA (93) 301 47 00

```
MAIN8   LDDR                    Carga las variables.
        RET
REPG    LD    A,^10             Produce el error  G  si  no  hay
        LD    BC,^0000          sitio para la nueva linea.
        JP    MAING
ENERR   LD    HL,(^5C5D)        CHADD.
        PUSH  HL                Lo guarda
        PUSH  DE                y el codigo de error (en E)
        CALL  ^16B0             Limpia el area de trabajo.
        LD    HL,ERRVAR         Hace  que  CHADD   apunte    al
        LD    (^5C5D),HL        comienzo de la palabra ERR.
        CALL  ^1C1F             Identifica  el  nombre  de    la
                                variable.
        POP   DE                Recupera el codigo de error.
        LD    A,E
        CP    ^15               Comprueba si se trata de  BREAK,
        JR    NZ,NBREAK         y salta si no.
WNOKEY  XOR   A
        IN    A,(^FE)           Lee todo el teclado de una vez.
        AND   ^1F
        CP    ^1F               Espera a que  no  este  apretada
        JR    NZ,WNOKEY         ninguna tecla.
        LD    A,^15             Pone otra vez el numero de error
NBREAK  CALL  MAIN9             Asigna OLDPPC y OSPPC.
        CALL  ^2D28             STACK-A. Coloca A en la pila.
        CALL  ^2AFF             LET. Pone el codigo de error  en
                                la variable basic err.
        POP   HL                Recupera  el  antiguo  valor  de
        LD    (^5C5D),HL        CHADD.
        POP   DE
        LD    A,^01             El  nuevo  valor  del  numero  de
        SUB   D                 instruccion debe ser uno mas que
        LD    (^5C44),A         el del ON ERR.
        LD    HL,LOERR          Coloca la direccion  de  retorno
        PUSH  HL                en la pila.
        LD    (IY+0),^FF        No hay error.
        JP    STMRET            Continua la ejecucion.
ERRVAR  DEFM  "ERR*"            Nombre de la variable err.
AUTOLI  DEFW  ^0000             Numero de linea.
AUTOIN  DEFW  ^0000             Incremento.
        DEFM  "ERR"             Otra vez el nombre.
AUTON   DEFB  ^00               Controla si estamos en AUTO.
MADD    LD    (^5C49),BC        Guarda el numero en E-PPC.
        LD    HL,(^5C5D)        Coloca la variable CH-ADD en los
        EX    DE,HL             registros DE.
        LD    HL,REPG           Pone la direccion de REPG en  la
        PUSH  HL                pila por si es necesario.
        LD    HL,(^5C61)        WORKSP.
        SCF                     Calcula la longitud de la  linea
        SBC   HL,DE             desde detras del numero hasta el
```

```
              LD    HL,^0001
              LD    (^5C16),HL     STRM6 STREAM 0 apunta a canal  K
              CALL  ^16B0          SET-MIN limpia todas  las  areas
                                   detras de E-LINE.
              RES   5,(IY+55)
              CALL  ^0D6E          Borra la parte baja de pantalla.
              CALL  CLCHAN         Arregla los canales de E/S.
              SET   5,(IY+2)       TV-FLAG
              POP   AF             Recupera el codigo del error.
              LD    B,A            Lo guarda en B.
              CP    ^0A            Si es mayor que 9 le suma 7 para
              JR    C,MAIN5        escribirlo bien.
              ADD   A,^07
      MAIN5   CALL  ^15EF          Escribe el codigo de error,
              LD    A,^20          y un espacio.
              RST   ^10
              LD    A,B
              LD    DE,^1391       Base para  la  tabla  con  los
                                   mensajes de error.
              CALL  ^0C0A          Escribe el mensaje,
              XOR   A
              LD    DE,^1536       seguido  de  una  coma  y  un
              CALL  ^0C0A          espacio.
              LD    BC,(^5C45)     PPC
              CALL  ^1A1B          Escribe el numero de linea.
              LD    A,^3A          Dos puntos
              RST   ^10
              LD    C,(IY+13)      SUBPPC
              LD    B,^00          Por ultimo escibe el  numero  de
              CALL  ^1A1B          la instruccion.
              CALL  ^1097          Limpia el area de edicion.
              LD    A,(IY+0)       El numero de error
              INC   A              lo incrementa
              CALL  NZ,MAIN9       Asigna OLDPPC Y OSPPC
              LD    (IY+10),^FF    NSPPC. Indica no salto.
              RES   3,(IY+1)       Modo K.
              JP    MAIN2
      MAIN9   CP    ^09            Es un STOP
              JR    Z,MAIN6
              CP    ^15            o BREAK en el programa?
              JR    NZ,MAIN7
      MAIN6   INC   (IY+13)        SUBPPC.  En  ese   caso   debe
                                   continuar  en   la   siguiente
                                   instruccion.
      MAIN7   LD    BC,^0003       Se prepara  para  dar  el  valor
              LD    DE,^5C70       correcto a las variables  OLDPPC
              LD    HL,^5C44       y OSPPC.
              BIT   7,(IY+10)      Si se iba a  producir  un  salto
              JR    Z,MAIN8        esta bien.
              ADD   HL,BC          Si no hay que usar PPC y SUBPPC.
```

```
            PUSH HL          fin de linea y lo guarda.
            LD   H,B         Pasa la longitud de la linea a
            LD   L,C         los registros HL y mira a ver si
            CALL ^196E       existe otra linea con ese numero
            JR   NZ,MADD1    Salta si no existe.
            CALL ^19B8       NEXT-ONE. Devuelve la direccion
                             de la siguiente linea.
            CALL ^19E8       RECLAIM2. Borra la linea.
     MADD1  POP  BC          Recupera la longitud de la nueva
            LD   A,C         linea y comprueba si se trata de
            DEC  A           un numero seguido de ENTER nada
            OR   B           mas.
            JR   Z,MADD2     Si es asi hemos acabado.
            PUSH BC          Guarda la longitud.
            INC  BC          Se necesitan cuatro bytes mas
            INC  BC          para codificar el numero de
            INC  BC          linea y la longitud.
            INC  BC
            DEC  HL          HL apunta a la direccion
                             anterior a la de destino.
            LD   DE,(^5C53)  Guarda la direccion PROG para
            PUSH DE          evitar problemas al introducir
                             la primera linea.
            CALL ^1655       MAKE-ROOM. Hace sitio para la
                             linea.
            POP  HL          Vuelve a colocar la direccion en
            LD   (^5C53),HL  la variable PROG.
            POP  BC          Recupera la longitud sin los
            PUSH BC          cuatro bytes iniciales.
            INC  DE          DE apunta al final del espacio
                             para la nueva linea.
            LD   HL,(^5C61)  WORKSP.
            DEC  HL          Le resta dos para que HL apunte
            DEC  HL          al final de la linea.
            LDDR             Copia la linea en el area de
                             programa.
            LD   HL,(^5C49)  Coge el numero de linea
            EX   DE,HL       y lo deja en DE.
            POP  BC          Recupera la longitud
            LD   (HL),B      y la coloca en su sitio, sobre
            DEC  HL          los bytes 3 y 4 de la linea.
            LD   (HL),C
            DEC  HL
            LD   (HL),E      Codifica de la misma forma el
            DEC  HL          numero de linea en los dos
            LD   (HL),D      primeros bytes.
     MADD2  POP  AF          Saca la direccion de REPG
            JP   MEXEC       y acaba volviendo a MEXEC.
```




• ***Organigrama del «nuevo» sistema operativo***

# SERVICIO DE EJEMPLARES ATRASADOS

Complete su colección de

# Todospectrum

**A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.**

**Núm. 1 • 250 pts.**
Cómo usar el microdrive/Programación Basic/Ampliación Basicare/Rutina despertador/Variables del sistema/Entrada datos mediante máscaras/Protección del software/Sintonice su Spectrum/Programas.

**Núm. 2 • 250 pts.**
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

**Núm. 3 • 250 pts.**
Novedades sonimag '84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

**Núm. 4 • 250 pts.**
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad /Calendario/Pascal/Programas.

**Núm. 5 • 250 pts.**
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por línea/Juego de la vida/Pascal/Así hacemos las portadas/Control de evaluaciones/Programas.

**Núm. 6 • 250 pts.**
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: tortugas para todos/Interrupciones del Z-80/Programas.







# DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE

# Todospectrum

**SIN NECESIDAD DE ENCUADERNACION**




(cada tapa es para 6 ejemplares)

**Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:**

**Todospectrum** Bravo Murillo, 377
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM ........................................ al precio de 250 pts.

Por favor envíenme ......... tapas para la encuadernación de mis ejemplares de TODOSPECTRUM, al precio de 600 pts. más gastos de envío.

El importe lo abonaré
□ POR CHEQUE □ CONTRA REEMBOLSO □ CON MI TARJETA DE CREDITO □ AMERICAN EXPRESS □ VISA □ INTERBANK

Número de mi tarjeta: □□□□□□□□□□□□□□□□□□□□□□□□

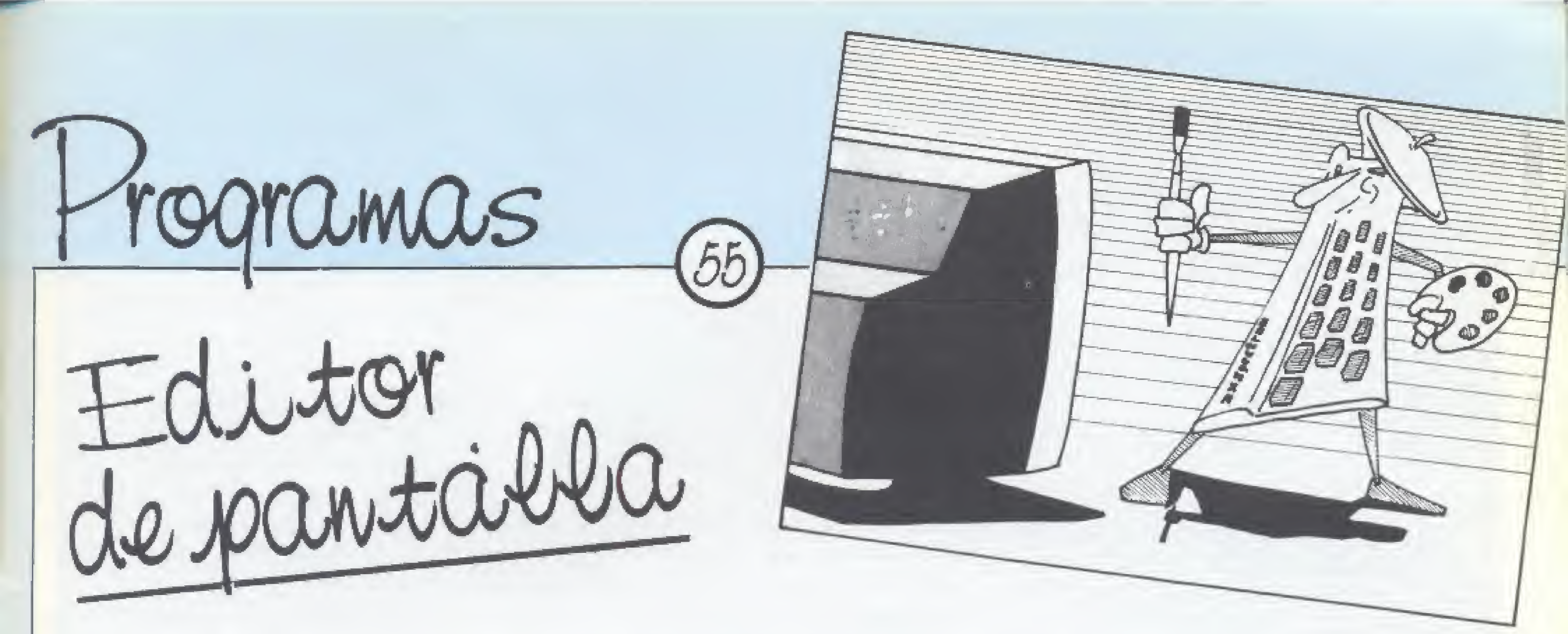
Fecha de caducidad .......................... Firma

NOMBRE ..................................................

DIRECCION ..................................................

CIUDAD .................................. C. P. ..................

PROVINCIA ..................................................



# Programas 55

## Editor de pantalla

El presente programa sirve para dibujar pantallas de presentación similares a las que aparecen al cargar un programa desde el *cassette*, tal y como hacen en los juegos de ULTIMATE, por ejemplo.

La calidad de las pantallas que permite hacer este Editor es similar a la de los mejores juegos comerciales, siempre y cuando nos pasemos dibujando frente al televisor el tiempo suficiente. La misma portada de esta revista se hace con un programa similar a éste.

Para introducir el programa teclear primero el listado 1 y grabarlo con SAVE «Editor» LINE 1

A continuación haga NEW y teclee el listado 2:

```
5  CLEAR 36954
10 LET sum=0
20 FOR X=36955 TO 37719
30 INPUT «Codigo»; a
```

**Con este programa un poquito de imaginación y grandes dosis de paciencia, tú también puedes ser nuestro portadista.**

40 POKE x,a: LET sum=sum+a: PRINT x,a
50 NEXT x
60 IF sum=68169 THEN PRINT «Ok»: STOP
70 PRINT «Error»

Ejecútelo e introduzca los 765 códigos de la figura 3, si obtiene error repáselos, y si no, grábelos con SAVE «c» CODE 36955, 765, a continuación de lo anterior.

Al ejecutarse el programa (con LOAD»») aparece en pantalla el menú de opciones:

1. DIBUJAR.—Permite realizar en la pantalla nuestro dibujo utilizando un punto como cursor que va pintando por donde pasa; se mueve con O, P, Q, A para izquierda, derecha, arriba y abajo respectivamente. Si a la vez pulsamos SPACE, el cursor se desplaza sin dibujar nada, y pulsando CAPS SHIFT va borrando. En la parte inferior de la pantalla aparece la posición del cursor en alta y baja definición. Pulsando L la posición del cursor se toma como origen para las próximas líneas o arcos que tracemos. Con F se traza una recta desde el punto indicado previamente con L y la posición del cursor. Con S se traza un arco, donde se nos pide el ángulo, que debe ser dado en grados sexagesimales. Pulsando C se nos pide el radio del círculo que se traza con centro el cursor. Al apretar X, la pantalla se cuadricula según los cuadros de baja definición; para volver a la situación normal se pulsa de nuevo X. SIMBOL SHIFT+V tiene el efecto de borrar la pantalla. Con M se vuelve al menú, quedando la pantalla guardada en memoria.

2. TEXTOS.—Esta opción sirve para imprimir las letras y símbolos del Spectrum, bien en tamaño normal o gigante. En tamaño normal aparece un cursor que se guía con las flechas en el Plus o con CAPS SHIFT y 5, 6, 7, 8 en el Spectrum 48K; apretando una tecla aparece el símbolo correspondiente en la posición del cursor. Podemos dibujar números, minúsculas, mayúsculas (con CAPS SHIFT) y signos (con SIMBOL SHIFT); para poner © se pulsa I+SIMBOL SHIFT. Se vuelve al menú con ENTER. En tamaño gigante se nos pide el texto, que debe ser menor de 32 caracteres, y se coloca con unas rayitas que marcan el extremo superior e izquierdo del mismo. Las rayitas se mueven con O, P, Q y A según lo habitual, se sitúan pulsando ENTER.

3. GRAFICOS DEFINIDOS POR EL USUARIO.—En la opción de dibujar podemos imprimir en la pantalla cualquiera de los 21 GDU, que aparecen en la parte inferior, y también espacios. El cursor se mueve igual que en textos normales, se sale igualmente con ENTER. En la opción de crear podemos dibujar nuestros propios GDU. Elegimos el gráfico a modificar con las teclas con flechas a izquierda o derecha o bien con CAPS SHIFT + 5 u 8, según el tipo de ordenador. Pulsando S se saca el gráfico de la memoria al cuadrado que aparece en pantalla. Con M se mete el dibujo del cuadrado en la memoria correspondiente. Con E lo que hay dibujado se invierte como si lo mirásemos en un espejo

```
   0>REM **EDITOR DE PANTALLA**
         1985 E. Salvador

   1 CLEAR 36954: LOAD ""CODE :
POKE 23658,0
   5 DEF FN f(x)=x-256*INT (x/25
6): DEF FN g(x)=x-(INT ((x-22528
)/32))*32-22528
   7 GO TO 9000
  10 LET or=a(4): LET de=16384:
GO SUB 8060
  15 INPUT "Coord. iniciales ";a
;" ";b
  17 IF a>255 OR a<0 OR b<0 OR b
>175 THEN  BEEP .1,-20: GO TO 15
  20 LET m=0: LET a1=a: LET b1=b
: LET l1=a: LET l2=b
  30 IF m THEN  PLOT  INVERSE 1;
a1,b1: LET m=0: GO TO 50
  40 PLOT a,b
  50 LET a=a+(IN 57342=190 AND a
<255)-(IN 57342=189 AND a>0): LE
T b=b-(IN 65022=190 AND b>0)+(IN
 64510=190 AND b<175)
  60 PRINT #1;AT 1,0;"Pos.-";a;"
 ";b;"    ";AT 1,17;INT ((175-b)
/8);" ";INT (a/8);"  "
  65 LET a1=a: LET b1=b
  70 IF IN 32766=190 THEN  GO TO
 50
  80 IF IN 65278=190 THEN  LET m
=1: GO TO 30
 100 IF INKEY$="l" THEN  LET l1=
a: LET l2=b: GO TO 30
 105 IF INKEY$="x" THEN  RANDOMI
ZE USR 37348
 110 IF INKEY$="f" THEN  PLOT l1
,l2: DRAW a-l1,b-l2: GO TO 30
 115 IF INKEY$="s" THEN  INPUT "
Angulo? ";an: PLOT l1,l2: DRAW a
-l1,b-l2,PI*an/180: GO TO 30
 120 IF INKEY$="c" THEN  INPUT "
Radio ";r: IF a>=r AND b>=r AND
255-a>=r AND 175-b>=r THEN  CIRC
```

y con G se gira un cuarto de vuelta en sentido antihorario. Apretando D podemos dibujar nuestro gráfico; aparece un cursor en el cuadrado que pinta blanco con O, negro con P, y que avanza sin pintar con L, para salir se pulsa ENTER y ya podemos meterlo en memoria, girarlo o in-vertirlo y meterlo en otro gráfico, etc. Para volver al menú se pulsa Q.

4. COLOREAR.—Esta opción nos permite dar color a la pantalla que hemos dibujado previamente. Con O, P, Q y A podemos desplazar un cursor por toda la pantalla. Pulsando C se nos pide el código de colores con que queremos pintar. Se debe introducir un número de 4 cifras: la primera indica FLASH, y debe ser 0 ó 1, la segunda es el BRIGHT y debe ser igualmente 0 ó 1, la tercera cifra es el código del PAPER y debe ser un número del 0 al 7, y la cuarta cifra es el INK, también un número del 0 al 7. Podemos cambiar de códigos pulsando C cuantas veces queramos. Al pulsar R aparece en pantalla en grande el cuadro que ocupaba el cursor cuando pulsamos R, lo podemos modificar con O, P y L igual que se hacía para crear GDU, y al apretar ENTER se sitúa en pantalla en la posición del cursor. Para volver al menú se pulsa M.

5. LINEAS INFERIORES.—En este apartado podemos subir el tercio inferior de pantalla del archivo que indiquemos (véase apartado 6), al tercio medio de la pantalla, enmarcado por dos líneas negras arriba y abajo. Posteriormente podemos bajar nuevamente este tercio a su lugar original. Gracias a esto, después de subirlas, podemos dibujar, colorear, etc., las dos líneas inferiores de la pantalla, que normalmente no son accesibles.

6. ARCHIVOS.—En la memoria del ordenador existen cuatro zonas que pueden albergar toda la pantalla. Las tres inferiores se denominan archivos 1, 2 y 3, y en ellos podemos meter la pantalla actual para luego comparar cuál es mejor, si va bien un dibujo o no, etc. También se puede sacar un archivo a pantalla. La cuarta zona o archivo se utiliza para guardar la pantalla en curso cuando volvemos al menú, creamos un gráfico, etcétera.

7. GRABAR.—Nos permite grabar en cinta la pantalla o uno de los tres archivos, bien en su posición de memoria o sacados previamente a la pantalla.

8. CARGAR.—Carga una pantalla en la posición en que la grabamos.

9. VERIFICAR.—Verifica lo que hemos grabado si era un archivo, la pantalla no se puede verificar porque aparece en ésta la inscripción «Bytes:» y, por tanto, no es igual que cuando la grabamos.

NOTA: La memoria requerida es de 48K. Como el programa está escrito en un Spectrum Plus, es necesario una modificación para el Spectrum normal: se deben cambiar los 190 y 189 por 254 y 253, respectivamente, en las líneas 50, 70, 80, 4560 y 4580. Para grabar el propio programa una vez introducido se teclea GOTO 9998.

**Eduardo Salvador Martínez**

```
LE a,b,r: GO TO 30
 125 IF INKEY$="/" THEN  CLS : G
O TO 30
 130 IF INKEY$="m" THEN  LET or=
16384: LET de=a(4): GO SUB 8060:
 RETURN
 140 GO TO 30
1510 CLS : PRINT "Normal (1) o g
igante (2)"
1515 IF INKEY$<>"" THEN  GO TO 1
515
1520 LET a$=INKEY$: IF a$<>"1" A
ND a$<>"2" THEN  GO TO 1520
1530 LET or=a(4): LET de=16384:
GO SUB 8060
1540 GO SUB VAL a$*500+1100
1550 LET or=16384: LET de=a(4):
GO SUB 8060
1560 RETURN
1600 LET a=0: LET b=a
1610 LET at=ATTR (a,b)
1615 LET a1=a: LET b1=b
1620 POKE 22528+b+a*32,214: LET
a$=INKEY$
1630 IF a$=CHR$ 8 THEN  LET b=b-
(b>0): GO TO 1680
1635 IF a$="AT " THEN  LET a$=
""
1640 IF a$=CHR$ 9 THEN  LET b=b+
(b<31): GO TO 1680
1650 IF a$=CHR$ 10 THEN  LET a=a
+(a<21): GO TO 1680
1660 IF a$=CHR$ 11 THEN  LET a=a
-(a>0): GO TO 1680
1665 IF a$=CHR$ 13 THEN  POKE 22
528+b1+a1*32,at: RETURN
1670 PRINT AT a,b;a$: IF a$<>""
THEN  LET b=b+1: IF b=32 THEN  L
ET a=a+1: LET b=0: IF a=22 THEN
 LET a=0
1680 POKE 22528+b1+a1*32,at: GO
TO 1610
2100 INPUT "Texto ";z$: LET lz=L
EN z$: IF LEN z$>32 THEN  BEEP .
```

# Programas

55

```
2,-20: GO TO 2100
2105 IF INKEY$<>"" THEN  GO TO 2
105
2110 LET yy=80
2120 PLOT  OVER 1;0,175-yy: DRAW
  OVER 1;6,0
2125 LET yy1=yy
2130 LET yy=yy+(INKEY$="a")-(INK
EY$="q")
2132 IF yy=175 THEN  LET yy=0
2134 IF yy=-1 THEN  LET yy=175
2140 PLOT  OVER 1;0,175-yy1: DRA
W  OVER 1;6,0
2150 IF INKEY$<>CHR$ 13 THEN  GO
 TO 2120
2160 INPUT "Ampliacion altura ";
ys: IF yy+8*ys>175 THEN  BEEP .2
,-20: GO TO 2160
2170 PRINT #1;AT 1,0;"Condicione
s longitud (s/n)": LET a$=INKEY$
: IF a$<>"s" AND a$<>"n" THEN  G
O TO 2170
2180 IF a$="n" THEN  LET xs=INT
(32/lz): LET xx=(256-8*xs*lz)/2:
 GO TO 2300
2185 IF INKEY$<>"" THEN  GO TO 2
185
2190 LET xx=120
2200 PLOT  OVER 1;xx,0: DRAW  OV
ER 1;0,6
2205 LET xx1=xx
2210 LET xx=xx+(INKEY$="p")-(INK
EY$="o")
2212 IF xx=256 THEN  LET xx=0
2214 IF xx=-1 THEN  LET xx=255
2220 PLOT  OVER 1;xx1,0: DRAW  O
VER 1;0,6
2230 IF INKEY$<>CHR$ 13 THEN  GO
 TO 2200
2240 INPUT "Ampliacion longitud
";xs: IF xx+xs*8*lz>255 THEN  BE
EP .2,-20: GO TO 2240
2300 LET i=23306: POKE i,xx: POK
E i+1,yy: POKE i+2,xs: POKE i+3,
ys: POKE i+4,8: LET i=i+4: FOR x
=1 TO lz: POKE i+x,CODE z$(x): N
EXT x: POKE i+lz+1,255
2310 RANDOMIZE USR 37408: RETURN

3010 CLS : PRINT "Dibujar (1) o
crear (2)"
3020 LET a$=INKEY$: IF a$<>"1" A
ND a$<>"2" THEN  GO TO 3020
3030 IF a$="2" THEN  GO TO 3500
3040 LET de=16384: LET or=a(4):
GO SUB 8060
3050 REM Una de las filas de let
ras es en modo graficos G: PRINT
 #1;AT 0,0;"     ABCDEFGHIJKLMNO
PQRSTU        ABCDEFGHIJKLMNO
PQRSTU       "
3060 LET a=0: LET b=a
3070 LET at=ATTR (a,b)
3080 LET a1=a: LET b1=b
3090 POKE 22528+b+a*32,214: LET
a$=INKEY$
3100 IF a$=CHR$ 8 THEN  LET b=b-
(b>0): GO TO 3180
3110 IF a$=CHR$ 9 THEN  LET b=b+
(b<31): GO TO 3180
3120 IF a$=CHR$ 10 THEN  LET a=a
+(a<21): GO TO 3180
3125 IF a$=" " THEN  PRINT AT a,
b;" ": GO TO 3170
3130 IF a$=CHR$ 11 THEN  LET a=a
-(a>0): GO TO 3180
3140 IF a$=CHR$ 13 THEN  POKE 22
528+b1+a1*32,at: LET or=16384: L
ET de=a(4): GO SUB 8060: RETURN
3150 IF a$<"a" OR a$>"u" THEN  G
O TO 3090
3160 PRINT AT a,b;CHR$ (CODE a$+
47)
3170 LET b=b+1: IF b=32 THEN  LE
T a=a+1: LET b=0: IF a=22 THEN
LET a=0
3180 POKE 22528+b1+a1*32,at: GO
TO 3070
3500 CLS : REM La fila inferior
de letras es en modo de graficos
 G: PRINT AT 2,5;"ABCDEFGHIJKLMN
OPQRSTU";AT 4,5;"ABCDEFGHIJKL
MNOPQRSTU"
3510 PLOT 94,46: DRAW 68,0: DRAW
 0,68: DRAW -68,0: DRAW 0,-68
3520 LET g=0
3530 POKE 37111,g: PRINT  OVER 1
; PAPER 5;AT 2,5+g;" ";AT 3,5+g;
" ";AT 4,5+g;" "
3540 LET a$=INKEY$: IF a$=CHR$ 8
 THEN  PRINT  OVER 1; PAPER 7;AT
 2,5+g;" ";AT 3,5+g;" ";AT 4,5+g
```

```
;" ": LET g=g-1: IF g=-1 THEN  L
ET g=20
3550 IF a$=CHR$ 9 THEN  PRINT  O
VER 1; PAPER 7;AT 2,5+g;" ";AT 3
,5+g;" ";AT 4,5+g;" ": LET g=g+1
: IF g=21 THEN  LET g=0
3560 IF a$="d" THEN  GO SUB 3650
3570 IF a$="e" THEN  RANDOMIZE U
SR 37217
3580 IF a$="g" THEN  RANDOMIZE U
SR 37276
3590 IF a$="s" THEN  FOR x=8 TO
15: PRINT  PAPER 7;AT x,12;"
    ": NEXT x: RANDOMIZE USR 371
12
3600 IF a$="m" THEN  RANDOMIZE U
SR 37164: PRINT AT 4,5+g;CHR$ (1
44+g)
3610 IF a$="q" THEN  RETURN
3620 GO TO 3530
3650 LET b=8: LET c=12
3660 LET at=ATTR (b,c): POKE 225
28+32*b+c,16+24*(at=40 OR at=56)
3670 IF INKEY$="o" THEN  POKE 22
528+32*b+c,56: LET c=c+1
3680 IF INKEY$="p" THEN  POKE 22
528+32*b+c,0: LET c=c+1
3690 IF INKEY$="1" THEN  POKE 22
528+32*b+c,56*(at=56 OR at=40):
LET c=c+1
3700 IF INKEY$=CHR$ 13 THEN  POK
E 22528+32*b+c,56*(at=40 OR at=5
6): RETURN
3710 IF c=20 THEN  LET c=12: LET
 b=b+1: IF b=16 THEN  LET b=8
3720 GO TO 3660
4510 LET or=a(4): LET de=16384:
GO SUB 8060
4515 INPUT "Coor. iniciales ";a;
" ";b: IF a>21 OR a<0 OR b<0 OR
b>31 THEN  BEEP .2,-20: GO TO 45
15
4520 LET a=22528+b+32*a: LET a1=
a: LET at=56: LET at1=at
4530 POKE a1,at: GO TO 4550
4540 POKE a1,at1
4550 LET a1=a: LET at1=PEEK a: P
OKE a,214
4560 LET a=a+(IN 57342=190 AND a
<23231)-(IN 57342=189 AND a>2252
8)+32*(IN 65022=190 AND a<23200)
-32*(IN 64510=190 AND a>22559)
4570 PRINT #1;AT 1,0;"Pos.- ";IN
T ((a-22528)/32);" ";FN g(a);"
"
4580 IF IN 32766=190 THEN  GO TO
 4530
4590 IF INKEY$="c" THEN  INPUT "
Codigo ";a$: IF LEN a$=4 THEN  G
O TO 4630
4600 IF INKEY$="r" THEN  GO SUB
5000
4610 IF INKEY$="m" THEN  POKE a1
,at1: LET or=16384: LET de=a(4):
 GO SUB 8060: RETURN
4620 GO TO 4540
4630 LET at=128*VAL a$(1)+64*VAL
 a$(2)+8*VAL a$(3)+VAL a$(4): GO
 TO 4540
5000 LET or=16384: LET de=a(4):
GO SUB 8060
5010 CLS : PLOT 94,46: DRAW 68,0
: DRAW 0,68: DRAW -68,0: DRAW 0,
-68
5020 POKE 37001,FN f(a): POKE 37
002,INT (a/256): RANDOMIZE USR 3
7000
5030 LET b=8: LET c=12
5040 POKE 22528+32*b+c,16+24*(AT
TR (b,c)=56 OR ATTR (b,c)=40)
5050 LET a$=INKEY$: IF a$=CHR$ 1
3 THEN  POKE 22528+32*b+c,56*(AT
TR (b,c)=40): RANDOMIZE USR 3705
5: LET or=a(4): LET de=16384: GO
 SUB 8060: RETURN
5060 IF a$="p" THEN  POKE 22528+
32*b+c,0: LET c=c+1
5070 IF a$="o" THEN  POKE 22528+
32*b+c,56: LET c=c+1
5075 IF a$="1" THEN  POKE 22528+
32*b+c,56*(ATTR (b,c)=56 OR ATTR
 (b,c)=40): LET c=c+1
5080 IF c=20 THEN  LET c=12: LET
 b=b+1: IF b=16 THEN  LET b=8
5090 GO TO 5040
6010 CLS : PRINT "Subir (1) o ba
jar (2)"
6020 LET a$=INKEY$: IF a$<>"1" A
ND a$<>"2" THEN  GO TO 6020
6030 IF INKEY$<>"" THEN  GO TO 6
030
6040 PRINT ''"Numero de archivo
```

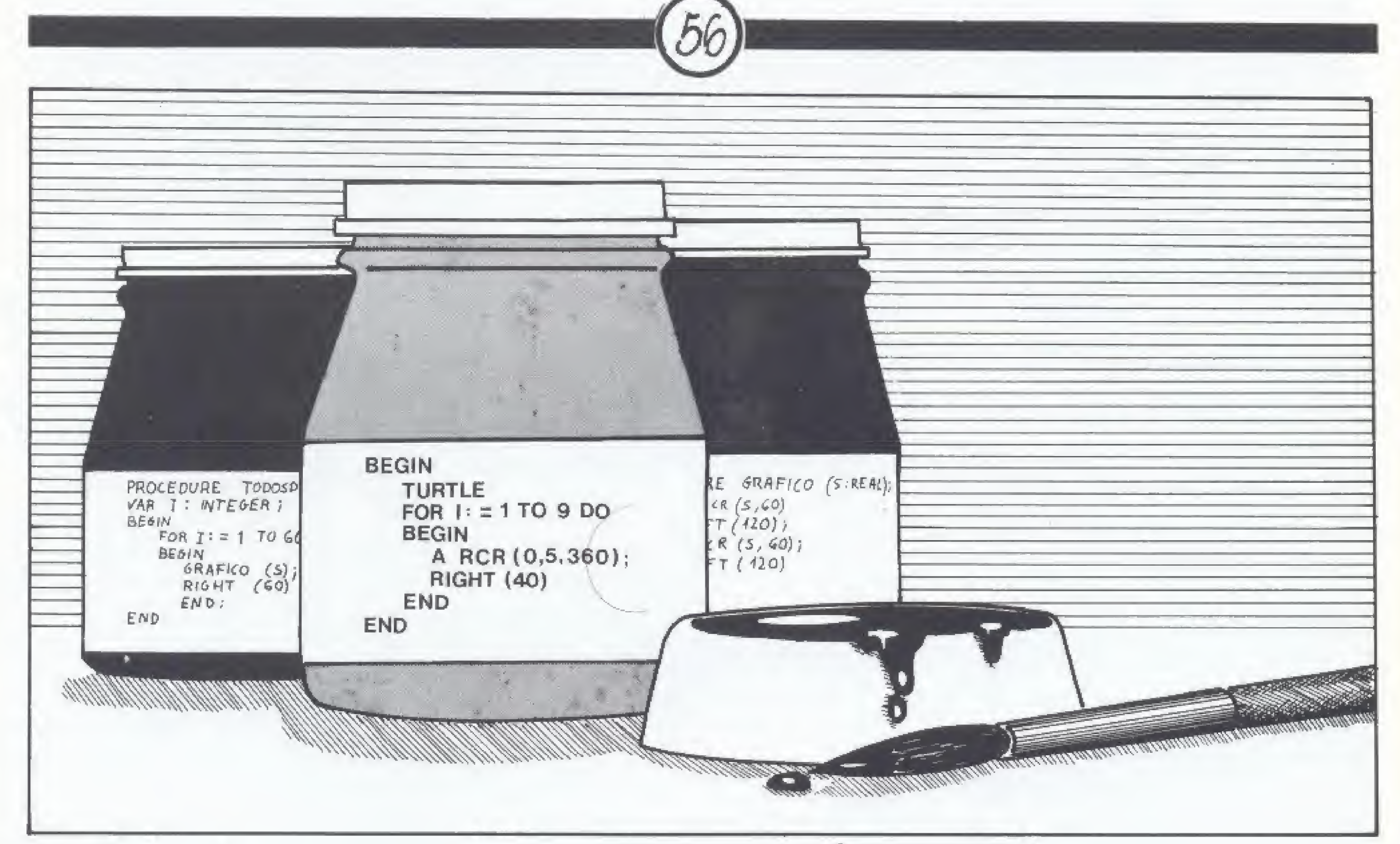
```
(1,2,3)"
6050 LET b$=INKEY$: IF b$<"1" OR
 b$>"3" THEN  GO TO 6050
6060 IF a$="1" THEN  LET or=a(VA
L b$)+4096: LET de=a(4)+2048
6070 IF a$="2" THEN  LET de=a(VA
L b$)+4096: LET or=a(4)+2048
6080 POKE 36956,FN f(or): POKE 3
6957,INT (or/256): POKE 36959,FN
 f(de): POKE 36960,INT (de/256)
6090 RANDOMIZE USR 36955
6100 FOR x=0 TO 31: POKE 60472+x
,255: POKE 62552+x,255: NEXT x
6110 RETURN
8000 CLS : PRINT TAB 5;"Meter(1)
 o sacar(2)"
8010 LET a$=INKEY$: IF a$<>"1" A
ND a$<>"2" THEN  GO TO 8010
8015 IF INKEY$<>"" THEN  GO TO 8
015
8020 PRINT ''TAB 5;"Numero archi
vo (1,2,3)"
8030 LET b$=INKEY$: IF b$<"1" OR
 b$>"3" THEN  GO TO 8030
8040 IF a$="1" THEN  LET or=a(4)
: LET de=a(VAL b$)
8050 IF a$="2" THEN  LET de=a(4)
: LET or=a(VAL b$)
8060 POKE 37709,FN f(or): POKE 3
7710,INT (or/256)
8070 POKE 37712,FN f(de): POKE 3
7713,INT (de/256)
8080 RANDOMIZE USR 37708
8090 RETURN
8100 CLS : PRINT "Archivos (1,2,
3) o pantalla (4)"
8110 LET a$=INKEY$: IF a$<"1" OR
 a$>"4" THEN  GO TO 8110
8120 IF INKEY$<>"" THEN  GO TO 8
120
8130 PRINT ''"En memoria (1) o p
antalla (2)"
8140 LET b$=INKEY$: IF b$<>"1" A
ND b$<>"2" THEN  GO TO 8140
8150 INPUT "Nombre ";n$: IF LEN
n$>11 THEN  BEEP .2,-20: GO TO 8
150
8160 IF b$="2" OR a$="4" THEN  L
ET or=a(VAL a$): LET de=16384: G
O SUB 8060: SAVE n$SCREEN$ : RET
URN
8170 SAVE n$CODE a(VAL a$),6912
8180 RETURN
8200 CLS : INPUT "Nombre ";n$
8210 LOAD n$CODE
8215 LET or=16384: LET de=a(4):
GO SUB 8060
8220 RETURN
8300 CLS : INPUT "Nombre ";n$
8310 VERIFY n$CODE
8320 RETURN
9000 BORDER 4: INK 0: PAPER 7: B
RIGHT 0: OVER 0: INVERSE 0: CLS
: INK 8: PAPER 8: BRIGHT 8
9010 DIM a(4): LET a(1)=37720
9015 FOR x=2 TO 4: LET a(x)=a(1)
+6912*(x-1): NEXT x
9017 RANDOMIZE USR 37380
9020 CLS : PRINT AT 5,14;"MENU"'
';TAB 11;"1 Dibujar";TAB 11;"2 T
extos";TAB 11;"3 GDU";TAB 11;"4
Colorear";TAB 11;"5 Lineas inf."
;TAB 11;"6 Archivos";TAB 11;"7 G
rabar";TAB 11;"8 Cargar";TAB 11;
"9 Verificar"
9025 PRINT AT 1,5;"**EDITOR DE P
ANTALLA**";AT 2,4;"▀▀▀▀▀▀▀▀▀▀▀▀▀
▀▀▀▀▀▀▀▀▀▀";AT 19,6; 1985 Edu
 Salvador"
9030 LET a$=INKEY$: IF a$="" THE
N  GO TO 9030
9040 IF a$>="1" AND a$<"6" THEN
 GO SUB (VAL a$-1)*1500+10: GO T
O 9020
9050 IF a$>"5" AND a$<="9" THEN
 GO SUB (VAL a$-6)*100+8000
9060 GO TO 9020
9998 SAVE "Editor" LINE 1: SAVE
"c"CODE 36955,765
9999 VERIFY "": VERIFY ""CODE
```

```
   5 CLEAR 36954
  10 LET sum=0
  20 FOR x=36955 TO 37719
  30 INPUT "Codigo ";a
  40 POKE x,a: LET sum=sum+a: PR
INT x,a
  50 NEXT x
  60 IF sum=68169 THEN  PRINT "o
k": STOP
  70 PRINT "error"
```

| | | | | |
|---|---|---|---|---|
| 36955 | 33 | 88 | 190 | 17 |
| 36959 | 88 | 236 | 1 | 0 |
| 36963 | 8 | 229 | 213 | 237 |
| 36967 | 176 | 209 | 235 | 124 |
| 36971 | 214 | 229 | 56 | 6 |
| 36975 | 6 | 17 | 62 | 10 |
| 36979 | 24 | 4 | 6 | 10 |
| 36983 | 62 | 17 | 14 | 0 |
| 36987 | 9 | 235 | 225 | 71 |
| 36991 | 9 | 6 | 1 | 237 |
| 36995 | 176 | 201 | 0 | 0 |
| 36999 | 0 | 33 | 22 | 89 |
| 37003 | 124 | 230 | 3 | 203 |
| 37007 | 39 | 203 | 39 | 203 |
| 37011 | 39 | 198 | 64 | 103 |
| 37015 | 1 | 88 | 164 | 9 |
| 37019 | 6 | 8 | 126 | 229 |
| 37023 | 33 | 236 | 89 | 79 |
| 37027 | 125 | 197 | 5 | 214 |
| 37031 | 32 | 16 | 252 | 111 |
| 37035 | 121 | 6 | 8 | 203 |
| 37039 | 127 | 40 | 2 | 54 |
| 37043 | 0 | 35 | 203 | 39 |
| 37047 | 16 | 245 | 193 | 225 |
| 37051 | 36 | 16 | 223 | 201 |
| 37055 | 42 | 137 | 144 | 124 |
| 37059 | 230 | 3 | 203 | 39 |
| 37063 | 203 | 39 | 203 | 39 |
| 37067 | 198 | 64 | 103 | 1 |
| 37071 | 88 | 164 | 9 | 6 |
| 37075 | 8 | 229 | 33 | 236 |
| 37079 | 89 | 125 | 197 | 5 |
| 37083 | 214 | 32 | 16 | 252 |
| 37087 | 111 | 14 | 0 | 6 |
| 37091 | 8 | 126 | 203 | 33 |
| 37095 | 167 | 32 | 2 | 203 |
| 37099 | 193 | 35 | 16 | 245 |
| 37103 | 121 | 193 | 225 | 119 |
| 37107 | 36 | 16 | 222 | 201 |
| 37111 | 14 | 33 | 88 | 255 |
| 37115 | 58 | 247 | 144 | 203 |
| 37119 | 39 | 203 | 39 | 203 |
| 37123 | 39 | 22 | 0 | 95 |
| 37127 | 25 | 6 | 8 | 126 |
| 37131 | 229 | 33 | 236 | 89 |
| 37135 | 79 | 125 | 197 | 5 |
| 37139 | 214 | 32 | 16 | 252 |
| 37143 | 111 | 121 | 6 | 8 |
| 37147 | 203 | 127 | 40 | 2 |
| 37151 | 54 | 0 | 35 | 203 |
| 37155 | 39 | 16 | 245 | 193 |
| 37159 | 225 | 35 | 16 | 223 |
| 37163 | 201 | 33 | 88 | 255 |
| 37167 | 58 | 247 | 144 | 203 |
| 37171 | 39 | 203 | 39 | 203 |
| 37175 | 39 | 22 | 0 | 95 |
| 37179 | 25 | 6 | 8 | 229 |
| 37183 | 33 | 236 | 89 | 125 |
| 37187 | 197 | 5 | 214 | 32 |
| 37191 | 16 | 252 | 111 | 14 |
| 37195 | 0 | 6 | 8 | 126 |
| 37199 | 203 | 33 | 167 | 32 |
| 37203 | 2 | 203 | 193 | 35 |
| 37207 | 16 | 245 | 121 | 193 |
| 37211 | 225 | 119 | 35 | 16 |
| 37215 | 222 | 201 | 6 | 8 |
| 37219 | 33 | 236 | 89 | 125 |
| 37223 | 197 | 5 | 214 | 32 |
| 37227 | 16 | 252 | 111 | 1 |
| 37231 | 0 | 8 | 203 | 33 |
| 37235 | 126 | 35 | 167 | 32 |
| 37239 | 2 | 203 | 193 | 16 |
| 37243 | 245 | 6 | 8 | 125 |
| 37247 | 144 | 111 | 203 | 65 |
| 37251 | 40 | 4 | 54 | 0 |
| 37255 | 24 | 2 | 54 | 56 |
| 37259 | 35 | 203 | 41 | 16 |
| 37263 | 241 | 193 | 16 | 208 |
| 37267 | 201 | 153 | 90 | 60 |
| 37271 | 24 | 60 | 90 | 90 |
| 37275 | 60 | 17 | 148 | 145 |
| 37279 | 6 | 8 | 33 | 12 |
| 37283 | 89 | 125 | 197 | 214 |
| 37287 | 32 | 16 | 252 | 111 |
| 37291 | 1 | 0 | 8 | 203 |
| 37295 | 33 | 126 | 35 | 167 |
| 37299 | 32 | 2 | 203 | 193 |
| 37303 | 16 | 245 | 121 | 18 |
| 37307 | 193 | 19 | 16 | 226 |
| 37311 | 17 | 148 | 145 | 6 |
| 37315 | 8 | 33 | 20 | 89 |
| 37319 | 197 | 45 | 16 | 253 |
| 37323 | 26 | 19 | 79 | 203 |
| 37327 | 65 | 40 | 4 | 54 |
| 37331 | 0 | 24 | 2 | 54 |
| 37335 | 56 | 203 | 41 | 125 |
| 37339 | 198 | 32 | 111 | 48 |
| 37343 | 238 | 193 | 16 | 225 |
| 37347 | 201 | 33 | 0 | 64 |
| 37351 | 78 | 124 | 6 | 5 |
| 37355 | 203 | 39 | 16 | 252 |
| 37359 | 167 | 121 | 40 | 4 |
| 37363 | 238 | 128 | 24 | 2 |
| 37367 | 238 | 255 | 119 | 35 |
| 37371 | 124 | 254 | 88 | 200 |
| 37375 | 24 | 230 | 0 | 0 |
| 37379 | 0 | 33 | 88 | 171 |
| 37383 | 17 | 89 | 171 | 54 |
| 37387 | 56 | 1 | 255 | 2 |
| 37391 | 229 | 237 | 176 | 225 |
| 37395 | 1 | 0 | 27 | 9 |
| 37399 | 216 | 84 | 93 | 19 |
| 37403 | 24 | 237 | 0 | 0 |
| 37407 | 0 | 33 | 15 | 91 |
| 37411 | 126 | 35 | 34 | 0 |
| 37415 | 91 | 111 | 60 | 200 |
| 37419 | 38 | 0 | 41 | 41 |
| 37423 | 41 | 237 | 75 | 54 |
| 37427 | 92 | 9 | 62 | 8 |
| 37431 | 50 | 4 | 91 | 58 |
| 37435 | 11 | 91 | 50 | 9 |
| 37439 | 91 | 58 | 10 | 91 |
| 37443 | 50 | 8 | 91 | 62 |
| 37447 | 9 | 50 | 5 | 91 |
| 37451 | 126 | 35 | 34 | 2 |
| 37455 | 91 | 7 | 50 | 6 |
| 37459 | 91 | 58 | 5 | 91 |
| 37463 | 61 | 32 | 50 | 58 |
| 37467 | 4 | 91 | 61 | 32 |
| 37471 | 24 | 58 | 14 | 91 |
| 37475 | 71 | 58 | 12 | 91 |
| 37479 | 79 | 58 | 10 | 91 |
| 37483 | 129 | 5 | 32 | 252 |
| 37487 | 50 | 10 | 91 | 42 |
| 37491 | 0 | 91 | 195 | 35 |
| 37495 | 146 | 50 | 4 | 91 |
| 37499 | 58 | 13 | 91 | 71 |
| 37503 | 58 | 9 | 91 | 128 |
| 37507 | 50 | 9 | 91 | 42 |
| 37511 | 2 | 91 | 195 | 64 |
| 37515 | 146 | 50 | 5 | 91 |
| 37519 | 58 | 12 | 91 | 71 |
| 37523 | 58 | 9 | 91 | 50 |
| 37527 | 7 | 91 | 58 | 13 |
| 37531 | 91 | 79 | 197 | 205 |
| 37535 | 196 | 146 | 193 | 58 |
| 37539 | 7 | 91 | 60 | 50 |
| 37543 | 7 | 91 | 13 | 32 |
| 37547 | 241 | 58 | 8 | 91 |
| 37551 | 60 | 50 | 8 | 91 |
| 37555 | 5 | 32 | 221 | 58 |
| 37559 | 6 | 91 | 195 | 80 |
| 37563 | 146 | 128 | 64 | 32 |
| 37567 | 16 | 8 | 4 | 2 |
| 37571 | 1 | 58 | 142 | 92 |
| 37575 | 238 | 255 | 71 | 58 |
| 37579 | 141 | 92 | 160 | 71 |
| 37583 | 58 | 8 | 91 | 230 |
| 37587 | 248 | 111 | 58 | 7 |
| 37591 | 91 | 254 | 192 | 208 |
| 37595 | 31 | 31 | 31 | 230 |
| 37599 | 31 | 103 | 203 | 28 |
| 37603 | 203 | 29 | 203 | 28 |
| 37607 | 203 | 29 | 203 | 28 |
| 37611 | 203 | 29 | 62 | 88 |

```
37615  180  103  58  142    37651  31  31  230  31     37687  0  0  0  0
37619  92  166  176  119    37655  181  111  235  33   37691  0  0  0  0
37623  58  7  91  71        37659  188  146  120  230  37695  0  0  0  0
37627  230  7  246  64      37663  7  79  6  0         37699  23  220  10  206
37631  103  120  31  31     37667  9  70  26  33       37703  11  231  80  26
37635  31  230  24  180     37671  6  91  203  70      37707  23  33  88  228
37639  103  120  23  23     37675  40  3  176  18      37711  17  0  64  1
37643  230  224  111  58    37679  201  47  176  47    37715  0  27  237  176
37647  8  91  71  31        37683  18  201  0  0       37719  201  0  0  0
```

56




# Subrutinas gráficas

Muchos de nuestros lectores opinan que el BASIC se queda corto y recurren a otros lenguajes. Este es el caso de Alberto Gómez Corona, que se ha inclinado hacia el Pascal y nos envía un conjunto de procedimientos y funciones que serán de gran utilidad a la hora de realizar gráficas.

Este conjunto de subrutinas emulan funciones del BASIC y otras añaden potentes comandos en tres dimensiones.

Como ejemplo se incluye un programa de dibujo de curvas cuadráticas. Estas pueden variar de forma cambiando simplemente el valor de las constantes a, b y c.

Los procedimientos son:

ATTRP (*ink, paper, bright, flash*). Cuyo manejo lo dan los nombres de los parámetros que hay que pasarle.

Aparte de éste y algún otro similar, los más interesantes son *plot3* (x, y, z) dibuja un punto con estas coordenadas teniendo en cuenta que (xc, yc, zc) está en el centro de la pantalla y que el punto de vista es (ax, ay, az).

*Draw3* (x, y, z) dibuja el punto igual que *plot3* pero uniéndolo con el punto dibujado anteriormente por medio de una recta. (Esto lo diferencia del *draw* del

BASIC que es relativo, mientras en éste se dan las coordenadas absolutas del punto).

Hay que decir que el programa se compiló con el Pascal de **Hisoft** y los resultados fueron muy precisos, aunque tarde algo en realizar el dibujo.

Por último nos queda hacer la advertencia de siempre referente a la «psicópata» de nuestra impresora, que el numeral lo cambia por un triángulo y el abrir llaves por dos puntos puestos horizontalmente en la parte superior del carácter. ¡Tened cuidado!

**Alberto Gómez Corona** **48K**

```
5C B4D3     5 PROGRAM REP3DIM;
B4D3     7 ¨alberto G. Corona 3-8
5}
B4D3    10 VAR XC,YC,ZC,AX,AY,AZ:
REAL;
B4DC    11 A,B,I,J,K:INTEGER;
B4DC    12 PROCEDURE ATTRP(I,P,B,
F:INTEGER);
B4DF    13 VAR ATR:INTEGER;
B4DF    14 BEGIN
B4F7    15  ATR:=I+P*8+B*64+F*128
;
B53C    16  POKE(23693,ORD(ATR));
B54B    17 END;
B556    18 FUNCTION INTERIOR(A,B:
INTEGER):BOOLEAN;
B559    19 BEGIN
B571    20   IF(A>=0)AND(A<=255)A
ND(B>=0)AND(B<=175)THEN
B5D0    21        INTERIOR:=TRUE
B5D3    22   ELSE INTERIOR:=FALSE
;
B5DC    23 END;
B5E6    73 PROCEDURE DRAW(X,Y:INT
EGER);
B5E9    74 BEGIN
B601    75   IF X<0 THEN POKE(232
96,ORD(^FF));
B624    76   IF X>=0 THEN POKE(23
296,ORD(1));
B649    77   IF Y<0 THEN POKE(232
97,ORD(^FF));
B66C    78 IF Y>=0 THEN POKE(2329
7,ORD(1));
B691    79 POKE(23728,ORD(ABS(X))
);
B6A3    80 POKE(23729,ORD(ABS(Y))
);
B6B5    81 INLINE(^D9,^E5,^D9,^ED
,^4B,^B0,^5C,^ED,^5B,^00,^5B,^CD
,^BA,^24,^D9,^E1,^D9);
B6C6    82 END;
B6D0    83
```

```
B6D0    84 PROCEDURE PLOT(X,Y:INT
EGER);
B6D3    85 BEGIN
B6EB    86 POKE(23728,ORD(X));
B6FA    87 POKE(23729,ORD(Y));
B709    88 INLINE(^ED,^4B,^B0,^5C
,^CD,^E5,^22);
B710    90 END;
B71A   100 PROCEDURE PLOT3(X,Y,Z:
REAL);
B71D   105 CONST XO=500; YO=128;Z
O=88;
B71D   110 VAR
B71D   120    R,R1,W,W1,V,V1:REAL;
B71D   125 X1,Y1,X2,Y2,X3,Y3,Z3:R
EAL;
B71D   130 A1,B1:INTEGER;
B71D   140
B71D   150 BEGIN
B735   170    R:=SQRT(SQR(XC-AX)+S
QR(YC-AY));
B77D   180    W:=(AY-YC)/R;
B7B1   190    V:=(XC-AX)/R;
B7E5   200    R1:=SQRT(SQR(R)+SQR(
AZ-ZC));
B822   210    W1:=AZ/R1; V1:=R/R1;
```



# Programas

56

```
B86F  220     X1:=X-AX;Y1:=Y-AY;
B8BF  230     X2:=X1*V-Y1*W;
B90E  240     Y2:=X1*W+Y1*V;
B959  250     Z3:=Z-AZ;
B981  260     Y3:=Y2;
B999  270     X3:=X2*V1-Z3*W1;
B9E8  280     Z3:=X2*W1+Z3*V1;
BA33  290
BA33  300     A1:=TRUNC(Y3*XO/X3+Y
O);
BA6F  310     B1:=TRUNC(Z3*XO/X3+Z
O);
BAAB  315     IF INTERIOR(A1,B1) T
HEN
BAC9  320     PLOT(A1,B1);
BAE0  325     A:=A1;B:=B1;
BAF2  340 END;
BB01  350 PROCEDURE DRAW3(X,Y,Z:
REAL);
BB04  360 VAR A1,B1:INTEGER;
BB04  370 BEGIN
BB1C  400          A1:=A;B1:=B;
BB2E  410          PLOT3(X,Y,Z);
BB61  420          IF INTERIOR(A,B)
 THEN BEGIN
BB79  425             DRAW(A1-A,B1-B
);
BBA8  427 END;
BBA8  430 END;
BBB4  500 FUNCTION CUADRICA(Y,Z:
REAL):REAL;
BBB7  510 CONST A=50;B=65;C=100;
BBB7  520 VAR  RAIZ:REAL;
BBB7  530 BEGIN
BBCF  540   RAIZ:=(1+SQR(Y/A)-SQ
R(Z/B));
BC23  550 IF RAIZ>=0 THEN
BC48  560 CUADRICA:=C*SQRT(RAIZ)
BC5E  570 ELSE CUADRICA:=1000;
BC83  580 END;
BC8F  600 PROCEDURE EJES;
BC92  605 BEGIN
BCAA  610 PLOT3(0,0,0);DRAW3(50,
0,0);
BCEC  620 PLOT3(0,0,0);DRAW3(0,5
0,0);
BD2E  630 PLOT3(0,0,0);DRAW3(0,0
,50);
BD70  640 END;
BD76  700 BEGIN
BD7F  705 WRITE(CHR(12));WRITE(C
HR(21));WRITE(CHR(0));
BD94  710 XC:=0;YC:=0;ZC:=0;
BDBB  720 AX:=400;AY:=400;AZ:=40
0;
BDE2  734 ATTRF(7,0,0,0);
BDF7  736 EJES;
BDFC  740 FOR J:=-50 TO 50 DO
BE19  750   FOR I:=-50  TO 50 DO
BE39  755      BEGIN
BE3C  760      PLOT3(CUADRICA(3*
I,3*J),3*I,3*J);
BE98  765      PLOT3(-CUADRICA(3
*I,3*J),3*I,3*J);
BEFC  767 END;
BF02  770 END.
End Address: BF04
Run?C
```

***Señoras y señores, he aquí un pequeño avance de lo más significativo que podrán ver en el próximo número. Gracias por su atención.***

**— Introducción a la electrónica digital.**

**— Una utilidad para mejorar el editor del BASIC.**

**— Spectrum 128K. Un primer vistazo.**

**— QL: introducción al SuperBasic.**



# Preguntas y respuestas

P ¿Qué tipo de corriente necesita el Spectrum? ¿Cuánto consume? Lo pregunto porque tengo una televisión portátil que conecto, por la toma del mechero, a la batería del coche y quisiera hacer lo mismo con mi Spectrum 48K o alimentarlo con pilas para poder así llevármelo de camping.

**Juan Barceló**
**Madrid**

R **La tensión nominal de entrada en el Spectrum es de 9 voltios, por lo que si deseas conectarlo a la batería del coche necesitarás de un alimentador que transforme los 12 V de ésta. Por otra parte, si tu ordenador es de 48K consume del orden de 700 mA, por lo cual la alternativa de utilizar pilas queda descartada (las consumiría en muy poco tiempo), pero no así la de utilizar la batería, que debe funcionarte a la perfección. Esperamos que tu Spectrum disfrute de esos fines de semana al aire libre que estamos seguros se merece.**

---

P Desearía saber si el listado del MONOPOLY de la revista número 8 se puede almacenar en *microdrive* o en *cassette,* y si se puede almacenar en *cassette,* cómo puedo almacenar las matrices. La línea 4 no la puedo introducir en el ordenador, saliendo una interrogación después de la sentencia LOAD cuando trato de introducirla.

**José Gallego**
**Alaquas**

R **No hay ningún problema si quieres jugar al Monopoly y no dispones de *microdrive;* simplemente deberás suprimir la línea 4 y, para introducir las matrices, dimensionar las variables que vas a utilizar (bastará que hagas: DIM C$ (38): DIM Q (38,8): DIM U$ (27,2): DIM V (27,2): DIM K (38): DIM D$ (11)), ejecutar los programas de introducción que hay después de cada serie en los datos anexos e introducir los datos uno por uno en el orden que aparecen en la revista. Todo esto lo deberás hacer antes de salvar en cinta el programa, de forma que las matrices queden grabadas con éste. Recuerda que no debes nunca hacer RUN para ejecutar el programa (esto borraría las variables), utiliza para ello GOTO 5.**

---

P En el programa *Manic Miner,* los muñecos que se mueven independientemente del movimiento de nuestro protagonista tendrán alguna rutina en C/M que creo será de interés para los lectores de vuestra revista, por lo que sugiero lo publiquéis. De esta forma los programas que enviemos serán mucho más atractivos.

**Eduardo Casado**
**Zaragoza**

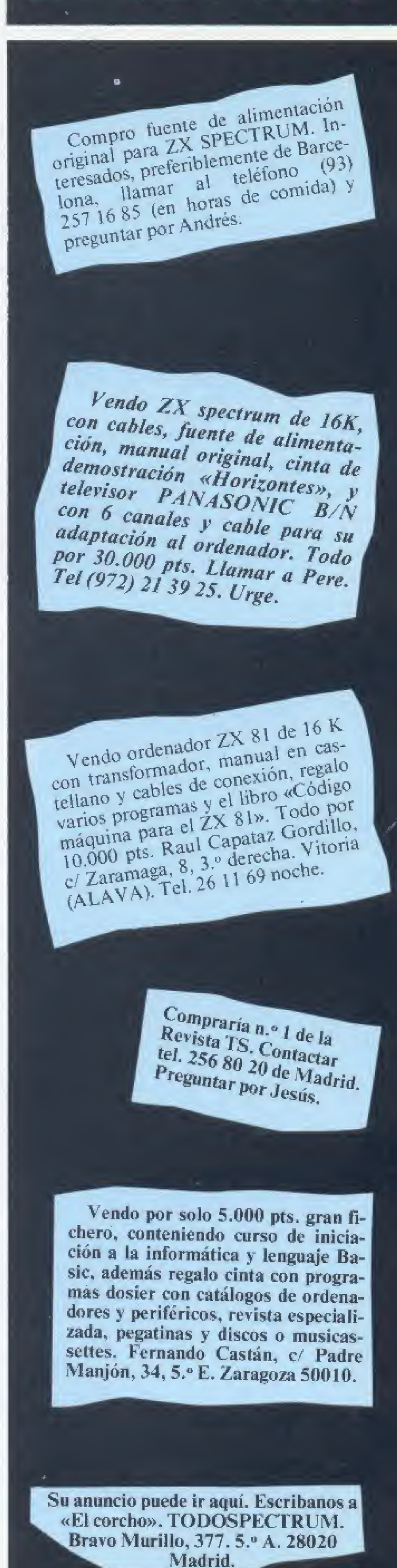
R **Nos gustaría mucho poder complacer tus deseos, pero las dificultades son muchas y variadas: en primer lugar debes tener en cuenta que el «genio» que programó el juego a que aludes tiene que dar de comer a sus hijos, no podemos publicar así como así un programa sin el permiso de su autor o la casa que lo distribuye. Por otra parte, el efecto que mencionas no es el resultado de una única rutina, sino de docenas de ellas de gran complejidad. Aparte del gran espacio que ocuparían en la revista, no creemos que haya muchos lectores que tengan el nivel necesario como para poder acoplarlas en sus programas. Por ahora deberás contentarte con cosas más sencillas, como las que mes a mes venimos ofreciendo.**

---

P En el magnífico artículo VARIABLES DEL SISTEMA, aparecido en el número 1 de vuestra revista, encuentro la falta de explicación a algunos *bits* que son importantes para el análisis de la ROM, y que tampoco están detallados en los libros habituales dedicados al código máquina.

Me interesaría conocer el signi-

# El corcho



**DIRECTOR:**
Simeón Cruz
**COORDINADOR EDITORIAL:**
Emiliano Juárez
**REDACCION:**
Enrique Larreta
Juan Arencibia, Fernando García, José C. Tomás, Luis M. Brugarolas, Ricardo García, Santiago Gala
**DISEÑO:** Ricardo Segura

Editado por
PUBLINFORMATICA, S. A.
**Presidente:** Fernando Bolín
**Director Editorial:** Norberto Gallego

**Administración:**
INFODIS, S. A.
**Gerente de Circulación y ventas:**
Luis Carrero
**Producción:**
Miguel Onieva
**Director de Marketing:**
Antonio González
**Servicio al cliente:**
Julia González. Tel. 733 79 69
**Administración:**
Miguel Atance
**Jefe de Publicidad**
María José Martín
**Dirección y redacción:**
Bravo Murillo, 377-5.º A. Tel. 733 74 13
Telex: 48877 OPZX e 28020 Madrid
**Administración y Publicidad:**
Bravo Murillo, 377-3 E. Tels. 733 96 62/96
**Publicidad Madrid:**
María Jose Martín
**Publicidad Barcelona:**
María del Carmen Ríos, Olga Martorell. Pelayo, 12.
Tel. (93) 318 02 89.
08001 Barcelona.
Depósito legal: M-29041-1984
Distribuye S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas-Madrid.
Fotomecánica: Karmat, C/ Pantoja, 10. Madrid.
Fotocomposición: Artecomp.
Imprime: Héroes, C/ Torrelara, 8. Madrid.

Distribuidor en VENEZUELA:
SIPAM, S.A.
AVD. REPUBLICA DOMINICANA, EDIF. FELTREC – OFICINA 4B BOLEITA SUR
CARACAS (VENEZUELA)
Esta publicación es miembro de la Asociación de Revistas de Información ARI asociada a la Federación Internacional de Prensa Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente TODOSPECTRUM)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377-5.º A
Tel. 733 74 13-28020 MADRID

Si deseas colaborar en TODOSPECTRUM remite tus artículos o programas a Bravo Murillo 377, 5.º A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.



# Preguntas y respuestas

ficado de cada uno de los *bits* de las variables siguientes:

23611 FLAGS
23612 TV FLAGS
23658 FLAGS 2
23665 FLAG X
23697 P FLAG

José García
Valencia

**R Estamos de acuerdo contigo: La falta de información referente a estas «banderas» es total, y un grave *hándicap* para quien pretende adentrarse en el desensamblado de la ROM. Hemos hecho un esfuerzo para ofrecerte unas pequeñas (dadas las limitaciones de esta sección) pero útiles notas del significado de cada uno de los *bits* utilizados; éstas se refieren a lo que indican cuando están activados (valor 1).**

**23611 5C3BH FLAGS:**
**bit 0: se ha imprimido un espacio.**
**bit 1: se está usando la impresora.**
**bit 2: modo «L».**
**bit 3: modo «K».**
**bit 5: una tecla ha sido pulsada.**
**bit 6: se está manejando una expresión numérica (0 si alfanumérica).**
**bit 7: se está ejecutando una orden (0 si se está comprobando su sintaxis).**
**23612 5C3CH TV FLAGS:**
**bit 0: utilizando parte baja de la pantalla (0 si parte principal).**
**bit 3: el cursor (K, L, C, etc.) ha sido cambiado.**
**bit 4: se está haciendo un listado automático (AUTOLIST).**
**bit 5: la parte baja de la pantalla necesita ser borrada.**
**23658 5C6AH FLAGS 2:**
**bit 0: la pantalla ha sido borrada.**
**bit 1: el buffer de la impresora ha sido usado.**
**bit 2: la cadena tratada está entre comillas.**
**bit 3: modo «C».**
**bit 4: usando canal «K».**
**23665 5CCDH FLAGX:**
**bit 0: la expresión tratada es una cadena simple (no dimensionada ni fragmentada).**
**bit 1: la variable tratada no existía.**
**bit 5: se está ejecutando un INPUT.**
**bit 6: el INPUT es numérico (0 si es alfanumérico).**
**bit 7: usando INPUT LINE.**
**23697 5C91H PFLAG:**
**bit 0 y 1: OVER 1.**
**bit 2 y 3: INVERSE 1.**
**bit 4 y 5: INK 9.**
**bit 6 y 7: PAPER 9.**
**23734 5CB6H FLAGS3:**
**bit 0: se está ejecutando un comando extendido.**
**bit 1: se está ejecutando CLEAR.**
**bit 2: ERRSP ha sido cambiado.**
**bit 3: utilizando la red local.**
**bit 4: ejecutando un LOAD.**
**bit 5: ejecutando un SAVE.**
**bit 6: ejecutando un MERGE.**
**bit 7: ejecutando un VERIFY.**

## Nota de la Redacción

A todos aquellos colaboradores voluntarios que nos envían programas para su publicación, queremos darles las gracias en público por su colaboración y os pedimos un pequeño favor; cuando nos enviéis programas para publicar, poner el nombre y demás datos de interés (dirección, teléfono, etcétera) en *todo lo que enviéis,* en la cinta, en el texto, en la carta, etcétera. Porque si lo ponéis sólo en el sobre corréis el riesgo de que os suceda lo que a:

*Jaime Melendi,* autor del programa de listas escolares del número de septiembre.

*Pablo Torrente,* creador del monitor de código máquina del número de agosto.

Que sólo escribieron su dirección en el sobre y por desgracia tiramos éste. Así pues, escribid la dirección en todos los sitios posibles y estos dos amigos que hemos comentado, por favor, enviadnos vuestra dirección o llamarnos por teléfono para que podamos pagaros vuestro trabajo publicado.

Gracias a todos por ayudarnos a hacer una revista mejor.

128K

# SPECTRUM 128 EL SUMMUM

Spectrum, como líder, marca un nuevo hito en la historia de los ordenadores familiares.

El Spectrum 128.

Gran capacidad de memoria. Teclado y mensajes en castellano, teclado independiente para operaciones numéricas y de tratamiento de textos...

Sinclair e Investrónica han desarrollado una auténtica novedad. En ningún lugar del mundo, salvo en los Distribuidores Exclusivos de Investrónica, podrás encontrar el nuevo Spectrum 128.

Sé el primero en tener lo último.

**SPECTRUM 128. NOVISIMUS**




**investronica**

Tomás Bretón, 62.
Tel. (91) 467 82 10.
Telex 23399 IYCO E.
28045 Madrid

Camp, 80.
Tels. (93) 211 26 58 - 211 27 54.
08022 Barcelona